



**Universitat Autònoma
de Barcelona**

Intranet pel HCR Cent Patins

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Sistemes
realitzat per
Albert Margarit Granero
i dirigit per
Daniel Blabia Girau

Escola d'Enginyeria

Sabadell, Setembre de 2010

El sotasignat, Daniel Blabia Girau,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en Albert Margarit Granero

I per a que consti firma la present.
Sabadell, Setembre de 2010

Signat: Daniel Blabia Girau

Titulació: Enginyeria Tècnica d'Informàtica de Sistemes

Projecte: Intranet pel HCR Cent Patins

Autor: Albert Margarit Granero

Director: Daniel Blabia Girau

Resum del projecte:

Intranet pel HCR Cent Patins és tracta d'un projecte de final de carrera d'Enginyeria Tècnica d'Informàtica de Sistemes a la UAB.

El projecte està estretament vinculat a l'entitat de Hockey Línia HCR Cent Patins de Rubí. Els objectius del projecte són desenvolupar una intranet de gestió econòmica pel Club respectant tots els procediments que s'executen al llarg d'una temporada a l'entitat.

En resum, en aquesta memòria hi ha tota la informació referent al projecte, estudi de la situació del Club, anàlisi dels requeriments de software i desenvolupaments tecnològics per al tractament de problemes de l'entitat.

ÍNDEX

1. Introducció	5
1.1. El HCR Cent Patins.....	6
1.2. La gestió del club esportiu	7
1.3. Estudi de la situació actual	9
1.3.1. Context.....	9
1.3.2. Lògica del sistema	10
1.3.3. Descripció física.....	10
1.3.4. Usuaris i personal del sistema.....	11
1.3.5. Diagnòstic del sistema	11
1.4. Objectius del projecte	12
1.4.1. Parts interessades	12
2. Requisits del projecte.....	14
2.1. Requisits funcionals	14
2.2. Requisits no funcionals	14
2.3. Restriccions del sistema	15
2.4. Catalogació i priorització dels requisits	15
3. Estudi de viabilitat.....	16
3.1. Alternatives i selecció de la solució.....	16
3.1.1. Alternativa 1.....	16
3.1.2. Alternativa 2.....	16
3.1.3. Alternativa 3.....	16
3.1.4. Solució proposada.....	17
3.2. Planificació del projecte	17
3.2.1. Recursos del projecte.....	18
3.2.2. Planificació detallada	19
3.2.3. Planificació temporal	20
3.2.4. Avaluació de riscos.....	21
3.2.5. Llista de riscos (R).....	21
3.2.6. Catalogació de riscos.....	22
3.2.7. Pla de contingència	22
3.3. Pressupost.....	22
3.3.1. Estimació cost de personal	22
3.3.2. Estimació cost de recursos.....	23
3.3.3. Resum i anàlisi cost benefici	23
3.4. Conclusions	23
4. El projecte Intranet pel HCR Cent Patins	24
4.1. Llenguatges informàtics	24
4.1.1. PHP.....	25
4.1.2. HTML.....	26
4.1.3. CSS.....	26
4.1.4. JavaScript	27
4.1.5. SQL	29
4.2. Eines tecnològiques	29

4.2.1.	Joomla 1.5	30
4.2.2.	MySQL	44
4.2.3.	XAMPP	46
4.3.	La metodologia del projecte	48
4.3.1.	Desenvolupament en cascada	48
5.	<i>Disseny de l'aplicació</i>	52
5.1.	Estudi del sistema	52
5.2.	Disseny del sistema	52
5.2.1.	Vista lògica	53
5.2.2.	Vista processos	64
5.2.3.	Vista desenvolupament	73
5.3.	Conclusions del disseny	75
6.	<i>Tests</i>	76
7.	<i>Conclusions</i>	78
7.1.	Futures modificacions i ampliacions	79
7.2.	Valoració personal	79
	<i>Referències</i>	80
	<i>ANNEX</i>	83

1. Introducció

Intranet pel HCR Cent Patins és un projecte de final que neix fruit de les inquietuds de l'autor vers el software i la voluntat d'ajudar a un club esportiu a solucionar problemes amb la gestió econòmica.

Destaquem que el caire d'aquest projecte no és purament tecnològic, sinó que també és social i econòmic.

L'aspecte tecnològic del projecte és crucial per la seva construcció donat que la nostra intenció es desenvolupar una aplicació que posteriorment esdevingui una eina tecnològica per al Club.

Hem buscat diferents alternatives i solucions per a solucionar els problemes del Club però ens hem decantat per buscar tecnologies amb llicències gratuïtes on poder desenvolupar una aplicació a mida. Fruit d'aquesta recerca hem topat amb Joomla Framework, aparentment un gestor de continguts com molts altres però, com veurem més endavant, sota aquesta aparença es troba un sistema molt robust estructurat amb classes i objectes que ens dona llibertat per desenvolupar eines i aplicacions complexes.

Joomla Framework ens proporciona molts dels requisits que cercàvem; seguretat, fiabilitat, potencia, etc. A més hem retallat despeses en llicències i hem après patrons arquitectònics desconeguts fins ara.

Parlem d'objectius de caràcter social donat que un dels principis d'aquest projecte és fomentar el treball en equip. El propòsit d'aquest software és descentralitzar les tasques de gestió del Club i aconseguir que tothom que hi estigui implicat realitzi petites feines amb l'objectiu d'aconseguir un control econòmic transparent. El nostre software vol ser el detonant dins de l'organització per crear un nou marc de treball.

Cal fer menció que el software anirà acompanyat d'una nova pàgina web creada pel Club que ha sigut dissenyada per un company de l'escola amb motiu de la realització del seu projecte final. Conseqüentment els dos projectes formen un de sol amb molt més potencial a nivell social.

Els propòsits econòmics són un dels pilars de l'aplicació. No aconseguirem un bon treball social si els objectius funcionals per a dur a terme una gestió segura no es compleixen. Aquesta finalitat és doncs crítica i desencadenant en gran part de les altres.

Per construir tant el nostre software de gestió com la pàgina web del Club ha calgut conviure a l'entorn de l'entitat durant un temps. En aquest període hem analitzat tots els departaments del club obtenint una visió global que ens ha permès tenir informació sobre procediments al Club i aprofundir en els problemes i errors que es detecten en aquests. Totes les solucions que s'han desenvolupat han esdevingut de la comunicació entre analistes i responsables de l'entitat.

1.1. El HCR Cent Patins



El Hockey Club Rubí Cent Patins és una entitat dedicada al món del Hockey Línia des de fa més de 12 anys.

A principis dels anys noranta un grup de patinadors en línia de la ciutat van sentir la inquietud de jugar a Hockey, un esport que arribava a Barcelona portat a mans d'americans que venien a gaudir dels Jocs Olímpics de Barcelona '92.

Aquesta febre pel Hockey Línia no va aparèixer només a Rubí, centenars d'aficionats es congregaven per jugar, inicialment, a carrers i places de Barcelona fins que als voltants del '95 les institucions i ajuntaments de diferents ciutats van facilitar instal·lacions per a començar la I Lliga Catalana de Hockey Línia. En aquesta primera competició el Cent Patins hi va participar juntament amb 12 altres Clubs d'arreu de Catalunya. Des d'aquell moment, el club rubinenc ha seguit la seva trajectòria d'expansió fins arribar avui dia a ser un referent a Catalunya i tot l'estat Espanyol.

Actualment els sènior femení i masculí militen a la primera divisió de Hockey Línia espanyola. El masculí absolut ha participat en quatre ocasions en competicions europees aconseguint al novembre de 2009 una Copa Confederació Europea. És l'únic equip espanyol en aconseguir l'or en una competició europea.

A les files dels equips sènior hi ha més de deu jugadors/es de la Selecció Nacional Absoluta.



A les categories base destaquen la participació de set equips en competicions catalanes, és a dir, un centenar de nens i nenes. A més de l'escola base, l'entitat organitza la Lliga d'Escoles de Hockey Línia de Rubí. La lliga la formen vuit equips de Rubí, tots ells procedents d'escoles de primària on els monitors de l'entitat imparteixen classes de patinatge i hockey en horari extraescolar. Enguany serà el tercer any d'activitat i s'espera augmentar un 25% les inscripcions de l'any anterior, unes vuitanta aproximadament.

Els èxits esportius i l'expansió del Club en els últims deu anys són clarament espectaculars.

En conseqüència a que el Club gestioni tal volum d'equips i de competicions, el numero de moviments econòmics i transaccions és elevat. No obstant, l'expansió esportiva del Club no ha estat acompanyada per una bona expansió administrativa, fet que ha produït mancances en l'organització. Posteriorment observarem el model actual de treball i analitzarem els seus punts febles.






1.2. La gestió del club esportiu

Les entitats esportives requereixen una gran dedicació, ordre i personal qualificat per tal d'aconseguir una gestió controlada i en bon estat tant administrativament com econòmicament. La comunicació, el treball en equip, la planificació i la previsió són factors essencials per mantenir una entitat amb garanties d'èxit. Lamentablement, les organitzacions esportives petites o d'esports minoritaris no solen tenir gent qualificada i professionals de l'administració a les seves files.







Per entendre la magnitud de les gestions que s'han d'efectuar en un club explicarem resumidament els tràmits que es fan al HCR Cent Patins en les seves diferents seccions.

Per començar hem d'identificar els ingressos i les despeses per tenir una concepció global de la procedència i el destí dels recursos del Club.










Ingressos:

-  Quotes jugadors.
-  Quotes de socis.
-  Esponsorització.
-  Subvencions públiques: procedents de la Generalitat i de l'Ajuntament de Rubí.
-  Organització d'activitats.

Despeses:

-  Federatives: Inscripció equips, fitxes jugadors, revisions mèdiques, arbitratges.
-  Treballadors club: entrenadors, monitors, directors esportius.
-  Material: equipament hockey, vestimenta, etc.
-  Viatges arreu d'Espanya i Europa dels equips Sènior.
-  Organització lligues escolars.
-  Assegurances.

Seccions esportives dins el Club:

-  Benjamins (2 equips)
-  Alevins (2 equips)
-  Infants
-  Juvenils
-  Junior
-  Sènior Masculí (2 equips, nacional i territorial)
-  Sènior Femení (2 equips, nacional i territorial)
-  Escola de patinatge per a adults
-  Escoles (gestió d'inscripcions de 5 escoles de Rubí)

Imaginem el gran treball administratiu que hi ha per tramitar tots els documents necessaris per rebre subvencions, inscriure equips i jugadors, organitzar lligues escolars, contractació de desplaçaments i hotels, etc.

Per fer-nos una idea, el Club té més de 180 nens entre alumnes d'activitats extraescolars i nens federats. A cada un d'ells li hem de tramitar una fitxa federativa i gestionar una revisió mèdica.

Cada una de les categories estan dirigides esportivament per els entrenadors i administrativament per delegats. Els tràmits específics els fa el departament de secretaria. Aquest i més departaments administratius pertanyen a l'equip de Junta Directiva. Està encapçalada un president i es seguit per vicepresidents, tesorero, secretari i vocals. Excloem d'aquest equip les persones encarregades de la part esportiva encara que si que formarien part de l'organigrama general del Club.

1.3. Estudi de la situació actual

En l'actualitat el HCR Cent Patins disposa d'un sistema molt poc pràctic per a la gestió de dades i el control de diners. Tot es basa en fulls de paper i en fulls d'Excel i Word.

D'altra banda la web del club és purament informativa i estàtica, sense interacció amb els usuaris.

1.3.1. Context

Si parlàvem dels èxits esportius de l'entitat amb gran orgull, no podem dir el mateix de l'administració. La ràpida expansió de l'esport a Rubí ha deixat enrere la gestió casolana que es feia als inicis i que encara perdura.

El Club no disposa d'una aplicació específica per saber l'estat de comptes amb fiabilitat. L'única manera de saber l'estat és mirant el compte resultat del banc, és a dir, si hi ha diners i un pressentiment d'èxit es produeix una transacció, sinó no.

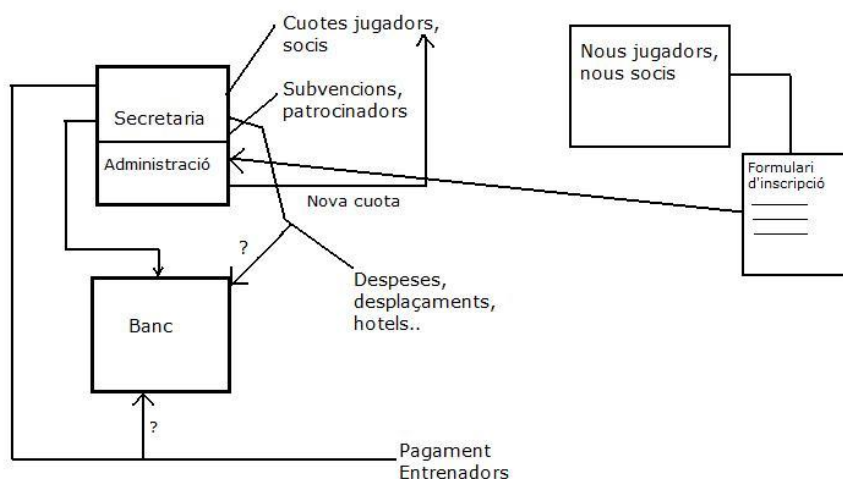
Aquest fet sumat al que no es fan pressupostos de temporada formals, escrits i documentats, condueixen a que l'estat del Club sempre es trobi en un estat precari, en que qualsevol incident que suposi un increment de les despeses inesperat per la secretaria pot suposar la defallida del sistema.

L'administració del Club està centralitzada a una persona que gestiona tots els tràmits tant administratius (federació, viatges, etc) com econòmics. Aquesta persona en moltes ocasions no pot finalitzar les tasques amb resultats positius ja que no disposa del temps necessari.

A més no hi ha un control de entrades i sortides detallat. Sovint les compres petites no tenen justificants. Al perdre aquests justificants no hi ha constància de que hagi hagut una sortida de diners, consegüentment les dades de despeses no són fiables al 100%.

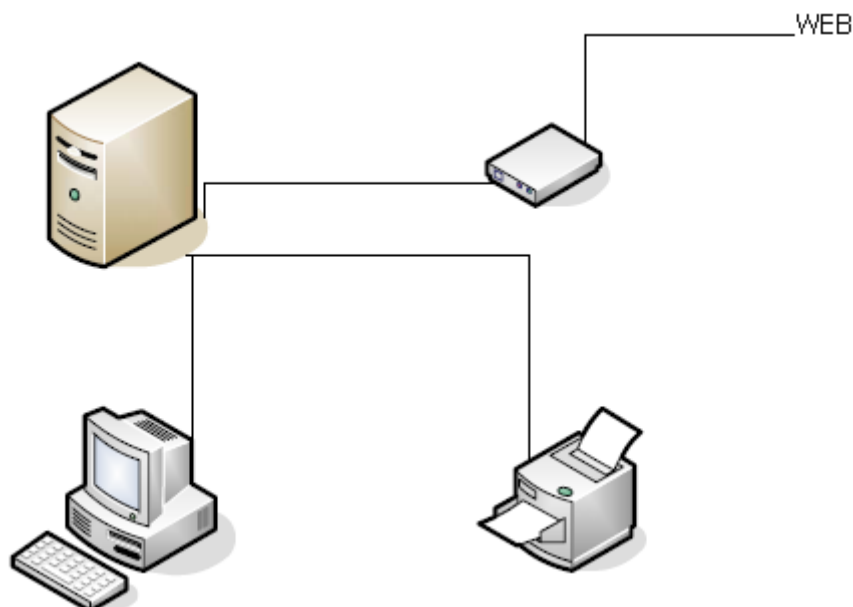
A més d'aquests factors interns, el Club no disposa d'un sistema de comunicació per a aficionats, jugadors i premsa preparat per a tal volum d'informació (pàgina web).

1.3.2. Lògica del sistema




El Club sempre es basa en l'estat de les comptes del banc. Abans de fer qualsevol operació consulta el saldo restant. No hi ha cap previsió anual ni mensual.

1.3.3. Descripció física



El sistema actual consta d'un ordinador bàsic per treballar amb fulls d' Excel i documents de textos i connexió a Internet per a la contractació de viatges, notificacions, comunicació amb federacions, etc. Està completament centralitzat a una terminal. No hi ha sistemes de recuperació de dades ni d'emmagatzematge de seguretat.

1.3.4. Usuaris i personal del sistema

<p>Secretaria/ Administració</p> 	<p>Gestiona la comptabilitat de l'entitat en general. Fa els cobraments a final de mes i realitza els pagaments de les diferents despeses originades per cada categoria. Altes i baixes de jugadors, socis. No hi ha personal disponible per assignar tasques.</p>
--	--

1.3.5. Diagnòstic del sistema

Deficiències: El sistema és controlat per un grup molt reduït de persones. La comptabilitat no es basa en cap previsió mensual ni anual, totes les operacions es fan depenent del saldo disponible.

Partint de que el Club té més de 180 jugadors i més de 100 socis, la comptabilitat hauria de ser sòlida i amb garanties de futur. Les tasques de les diferents seccions del club estan centralitzades, no hi han uns departaments físicament separats.

Al no tenir un sistema informatitzat i distribuït hi ha un gran risc de perdre informació i documents. A més hi ha més possibilitats de cometre errors administratius.

Tots els èxits esportius trontollen per la mala gestió del Club.

Millores: La feina ha de ser distribuïda en diferents seccions físicament separades i amb un organigrama establert i clar. La creació d'un espai virtual per a poder dipositar les dades de les despeses de manera multilateral, és a dir, on hi intervinguin varies persones encarregades d'alguna de les seccions del Club, seria molt més eficaç.

Necessitat de fer pressupostos reals i estudiats que es corresponguin amb el volum del Club.

Aconseguir un control econòmic estable per a les diferents categories a més d'un compte resultat total o parcial.

Informatitzar el sistema d'altres i baixes. Disposar d'una base de dades amb informació de tots els membres del club, oferint la possibilitat d'inserir nous membres o de esborrar-los. D'aquesta manera evitem pèrdues d'informació i evitem l'ús del paper com a principal documentació per a el departament administratiu.

1.4. Objectius del projecte

La nostra aplicació té com a objectiu global millorar el tractament de dades a nivell econòmic i fomentar la participació de tercers mitjançant la realització de petites tasques d'inserció i manteniment de dades d'alguna de les categories. Les premisses són suplir totes les mancances que té actualment el sistema amb transparència, seguretat i sobretot amb un sistema distribuït.

Els objectius crítics d'aquest projecte són:

- 🚦 Aconseguir una administració econòmica segura i transparent.[O1]
- 🚦 Tenir un compte resultat sempre actualitzat i enregistrat.[O2]
- 🚦 Obtenir un informe anual sobre l'estat econòmic amb un gràfic que ens indiqui els creixements i decreixements de l'entitat durant l'any.[O3]
- 🚦 Proporcionar a usuaris, socis i persones implicades una font d'informació fiable.[O4]
- 🚦 Facilitar la realització de pressupostos anuals[O5].
- 🚦 Tenir un històric de temporades amb tots els seus registres per aconseguir pressupostos més acurats.[O6]

Observem els objectius prioritaris:

- 🚦 Aconseguir una bona comunicació entre tots els departaments del Club.[O7]
- 🚦 Proporcionar eines per la generació i administració de continguts del web per a usuaris no familiaritzats amb llenguatges de programació.[O8]

1.4.1. Parts interessades

Stakeholders:

Responsable/s de l'entitat	Definició del projecte, aprovació del producte final, subvenció per la realització.
Director del Projecte	Supervisor del producte durant el seu desenvolupament i a la seva finalització.

Perfils d'usuaris:

Administrador del Sistema	Gestiona la informació del Sistema. És també el gestor de comptes d'usuari.
Gestors	Són els encarregats d'introduir les entrades i sortides de diners a l'aplicació generadora del compte resultat. Hi haurà un per cada categoria.

Delegats	Encarregats d'introduir les despeses d'arbitratge i aigua, de cada una de les categories, a la finalització de la jornada. També hauran d'inserir les estadístiques dels jugadors després de cada partit.
Altres	Terceres persones que puguin encarregar-se de la inserció de dades de despeses o ingressos d'algun concepte determinat.

Project Team

Cap de Projecte	Gestiona i planifica el projecte.
Analista	Fa un anàlisi de la situació del Club, defineix els requisits funcionals i no funcionals per a la futura implementació de l'aplicació.
Dissenyador Tècnic/Programador	Programa i desenvolupa l'aplicació web seguint l'anàlisi prèviament estudiat i definit.
Tècnic de proves	Realitza les proves pertinents amb l'objectiu de trobar i possibles errors a l'aplicació.
Director del Projecte o tutor	Supervisa la feina de l'alumne.

2. Requisits del projecte

El nostre projecte es basarà en la creació d'una aplicació web per així tenir el sistema distribuït que desitgem. Per tal de realitzar un projecte amb garanties hem de tenir molt clars quins seran els requisits funcionals així com implementar eines que ens ofereixin seguretat a la nostra base de dades. Tots aquests requisits hauran de ser especificats dins del marc legal vigent. A continuació definirem tots els requisits del nostre sistema.

2.1. Requisits funcionals

- a) Altes, baixes, modificacions de despeses per categoria.
- b) Altes, baixes, modificacions d'ingressos per categoria
- c) Control d'accés dels usuaris de l'aplicació.
- d) Mostra de dades de caràcter econòmic per categories.
- e) Gestió comptes club per temporades.
- f) Gràfics estadístics de l'estat de comptes.
- g) Informes detallats de l'estat de comptes anuals.
- h) Arxivador de temporades.
- i) Creació de pressupostos detallats amb despeses i ingressos.
- j) Flexibilitat per a inserir noves funcionalitats.
- k) Còpies de seguretat i recuperació de dades.

2.2. Requisits no funcionals

- a) Interfície d'usuari fàcil i interactiva.
- b) Normalització de la base de dades i accés segons l'estàndard SQL 99 (ISO/IEC 9075:1999).
- c) Els recursos utilitzats per l'aplicació han d'estar ajustats a la mida de l'entitat.
- d) Tolerància a errades i a accions incorrectes.
- e) Control de totes les entrades i sortides d'usuaris.
- f) El control d'accés es farà d'acord amb la llista d'usuaris i permisos.
- g) La seguretat de les dades: el servidor haurà de tenir restriccions d'accés físic.

2.3. Restriccions del sistema

- a) L'aplicació s'implementarà en un entorn Windows.
- b) L'aplicació ha d'adaptar-se al sistema físic disponible en l'entitat.
- c) El projecte ha d'estar finalitzat abans del 30 de setembre de 2010.
- d) L'aplicació s'ha de desenvolupar utilitzant programari lliure
- e) L'aplicació ha de suportar diferents exploradors d'Internet (IE7, Firefox, Safari)

2.4. Catalogació i priorització dels requisits

Prioritats dels Requisits funcionals

	A	B	C	D	E	F	G	H	I	J	K
Essencial	X	X	X	D	X				X	X	X
Condicional						X	X	X			
Opcional											

Prioritats dels Requisits no funcionals

	A	B	C	D	E	F	G
Essencial	X	X	X	X	X	X	
Condicional							X
Opcional							

Relació dels requisits amb els objectius del sistema:

[O1]: c,d,h,k.

[O2]: a,b,d,f

[O3]: f,g.

[O4]: c.

[O5]: i.

[O6]: g,h,k.

[O7]: 2ª Part projecte (Web pel HCR Cent Patins)

[O8]: 2ª Part projecte (Web pel HCR Cent Patins)

3. Estudi de viabilitat

3.1. Alternatives i selecció de la solució

3.1.1. Alternativa 1

Adquisició d'un software de gestió esportiva.

Per exemple: DEPORWIN (<http://www.deporwin.com>) de la companyia T-Innova.



Funcionalitats:

- Terminals d'autoservei.
- Facturació bancària segons C19.
- Formes de pagament parametitzables.
- Enviament de quotes automatitzat.
- Mòdul de seguretat personalitzable.

Cost: Desconegut.

3.1.2. Alternativa 2

Adaptació de l'aplicació Deporwin a la nostre entitat.

Característiques:

- Introducció de les dades econòmiques del Club a l'aplicació.
- Manteniment de l'aplicació a càrrec de l'empresa subministradora.
- Mòduls extres comprant-los directament a l'empresa subministradora.

Cost: Desconegut, però més elevat que la compra del producte.

3.1.3. Alternativa 3

Desenvolupament d'una aplicació web per a realitzar la gestió esportiva del Club.

Característiques:

- Ajustable als requisits reals i actuals del Club
- Ajustable als recursos disponibles per l'entitat.
- Gestió esportiva i web en un mateix entorn.
- Aplicació preparada per a futurs canvis i oberta a nous mòduls.

3.1.4. Solució proposada

	Cost Adquisició	Cost Adaptació	Nous Recursos	Suport	Nivell d'integració	Complexitat	Formació
A1	Des.	Des./Alt	Adaptable	Negociable	Baix	Baixa	Des.
A2	Des.	Des./Alt	No	Negociable	Baix	Mitjana	Des.
A3	0,00 €	Pressupost projecte	No	Inclòs al projecte	Alt	Alta	Inclusa al projecte

Justificació de la solució: A priori un programa de gestió esportiva ja dissenyat és una solució força econòmica per al nostre projecte. Hem de tenir en compte que la nostra entitat no només precisa d'un software de gestió, anteriorment hem explicat les mancances a nivell informatiu (web).

La situació del Club fa indispensable la necessitat de tenir una eina per a millorar la comunicació entre departaments, oferir bona informació a socis i seguidors i aconseguir una millora en la gestió econòmica.

Amb al desenvolupament del nostre i l'altre projecte paral·lel es crearà un entorn personalitzat i específic que garanteix tots els requisits actuals.

Intranet pel HCR Cent Patins vetllarà per una administració econòmica transparent i segura, Web pel HCR Cent Patins oferirà l'informació del Club amb una web atractiva i interactiva amb usuaris, socis, jugadors i seguidors.

Hem decidit crear aquesta aplicació a Internet conjunta perquè es ajustable, es potent i realitza les tasques on actualment es troben deficiències a l'entitat.

3.2. Planificació del projecte

Una de les parts important del cicle de vida d'un projecte amb metodologia en cascada (punt 4.3) és la planificació. En projectes on les dates d'entrega són molt ajustades per els usuaris finals (compradors) cal tenir una planificació clara i real.

Per estimar la duració tenim en compte no només el procediment tècnic a utilitzar en el projecte, sinó que també els recursos, costos, etc. A mesura que el software augmenta, creix la interdependència entre elements del software, si s'originen problemes, en ocasions, es poden arrossegar fins el final del cicle.

A continuació estimem i planifiquem el nostre projecte.

- ✚ Calendari del projecte: El projecte es desenvoluparà d'octubre 2009 al juliol del 2009 amb una dedicació de 15 hores setmanals. En total es dedicaran 500 hores al projecte.
- ✚ Data començament: 1 de Novembre 2009
- ✚ Data finalització: 1 juliol 2010
- ✚ Eines de planificació i control: Ms Project

3.2.1. Recursos del projecte

Recursos humans	Valoració
Director del projecte (D)	40€/h
Responsable del projecte(alumne) (A)	25€/h

Recursos materials: s'utilitzaran els disponibles en el club. S'utilitzarà programari de domini públic.

Costos indirectes: amortització dels recursos de desenvolupament.

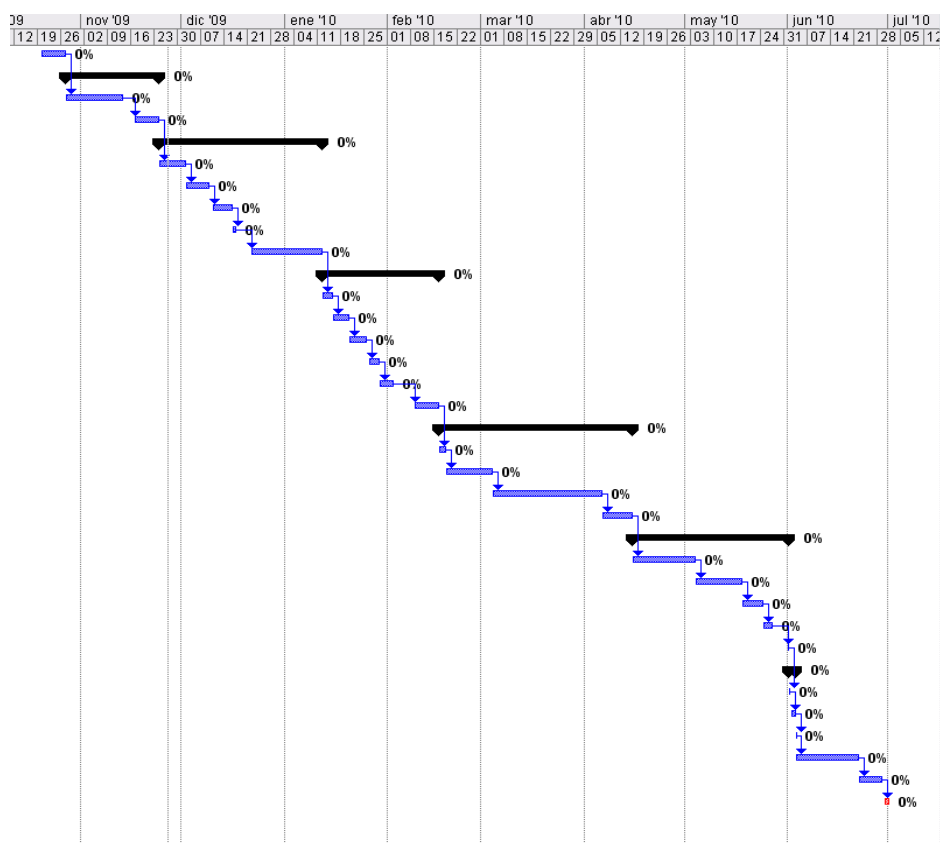
3.2.2. Planificació detallada

Nº	Descripció de l'activitat	Durada	Recursos	Pred.
1	Inici del projecte: assignació i matriculació del projecte	3h	A, D(50%)	
2	Planificació	40h		
3	Estudi de Viabilitat	38h	A	1
4	Aprovació estudi viabilitat	2h	A, D(50%)	3
5	Anàlisi de l'aplicació	50h		
6	Anàlisi de requisits (cassos d'ús)	20h	A	4
7	Anàlisi de dades (base de dades)	10h	A	6
8	Anàlisi de la seguretat i legalitat	13h	A	7
9	Documentació de l'anàlisi	5h	A	8
10	Aprovació de l'anàlisi (Punt de control)	2h	A, D(50%)	9
11	Disseny de l'aplicació	60h		
12	Disseny de la base de dades	10h	A	10
13	Disseny modular de l'aplicació	10h	A	12
14	Disseny de d'interfície, ajuda en línia	10h	A	13
15	Disseny de les proves (test)	10h	A	14
16	Documentació del disseny	7h	A	15
17	Aprovació del disseny (Punt de control)	2h	A, D(50%)	16
18	Desenvolupament de l'aplicació	170h		
19	Preparació entorn de desenvolupament	10h	A	17
20	Configuració base de dades	40h	A	19
21	Mòdul d'adquisició de dades i funcionalitats de l'aplicació	90h	A	20
22	Desenvolupament de l'interfície d'usuaris	30h	A	21
23	Test i proves	120h		
24	Proves unitàries	50h	A	22
25	Proves d'integració	40h	A	24
26	Proves d'estres (incidències, riscos)	19h	A	25
27	Documentació de desenvolupament i test	10h	A	26
28	Aprovació del desenvolupament i proves (Punt de control)	1h	A, D(50%)	27
29	Implantació	10h		
30	Instal·lació	3h	A	28
31	Proves reals	5h	A	30
32	Formació d'usuaris	2h	A	31
33	Generació de documents (memòria del projecte)	50h	A	32
34	Tancament del projecte	2h	A, D(50%)	33
35	Defensa del projecte	5h	A	34

3.2.3. Planificació temporal

Nombre de tarea	Duracion
Inici del projecte: assignació	3 horas
Planificació	19,88 días
Estudi Viabilitat	38 horas
Aprovació estudi viabilitat	2 horas
Anàlisi de l'aplicació	31,94 días
Anàlisi de requisits(cas	20 horas
Anàlisi de dades(base i	10 horas
Anàlisi de la seguretat i	13 horas
Documentació de l'anàl	5 horas
Aprovació de l'anàlisi (F	2 horas
Disseny de l'aplicació	24,94 días
Disseny de la base de d	10 horas
Disseny modular de l'ap	10 horas
Disseny de l'interfície, a	10 horas
Disseny de les proves (10 horas
Documentació del diss	7 horas
Aprovació del disseny (F	2 horas
Desenvolupament de l'aplic	39,25 días
Preparació entorn de de	10 horas
Configuració base de d	40 horas
Mòdul d'adquisició de d	90 horas
Desenvolupament de l'i	30 horas
Test i proves	32,81 días
Proves unitàries	50 horas
Proves d'integració	40 horas
Proves d'estress (incidi	19 horas
Documentació de dese	10 horas
Aprovació del desenvol	1 hora
Implantació	2,25 días
Instal·lació	3 horas
Proves reals	5 horas
Formació d'usuaris	2 horas
Generació de documents	50 horas
Tancament del projecte	2 horas
Defensa del projecte	5 horas

Durada del projecte: 500h.



3.2.4. Avaluació de riscos

L'avaluació de riscos és sempre una de les parts més importants abans de l'inici d'un projecte. S'han de definir bé quins riscos podem trobar al llarg del cicle.

3.2.5. Llista de riscos (R)

R1. Planificació temporal optimista: estudi de viabilitat. No s'acaba en la data prevista, augmenten els recursos.

R2. Manca alguna tasca necessària: estudi de viabilitat. No es compleixen els objectius del projecte.

R3. Equip del projecte massa reduït: estudi de viabilitat. Endarreriment en la finalització del projecte, no es compleixen els objectius del projecte.

R4. Equip del projecte massa ampli: estudi de viabilitat. Finalització del projecte abans del previst, pèrdua econòmica.

R5. Eines de desenvolupament inadequades: implementació. Endarreriment en la finalització del projecte, menys qualitat, etc.

R6. Dificultat per accedir als stakeholders: estudi de viabilitat, anàlisi, proves, formació. Manquen requisits o són inadequats, endarreriments, insatisfacció usuaris.

R7. No es fa correctament la fase de proves: desenvolupament, implantació. Manca de qualitat, deficiències en l'operativa, insatisfacció usuaris, pèrdua econòmica.

R8. Incompliment d'alguna norma, reglament o legislació: en qualsevol fase. No es compleixen els objectius, repercussions legals.

R9. Manca d'implantació de mesures de seguretat: estudi de viabilitat, anàlisi, desenvolupament. Pèrdua d'informació, incompliment legal, pèrdues econòmiques.

R10. Abandonament d'algun membre de l'equip: en qualsevol fase. Endarreriment en la finalització del projecte, augment d'hores de feina pels membres del grup.

R11. Abandonament del projecte abans de la finalització: en qualsevol fase. Pèrdues econòmiques, frustració.

3.2.6. Catalogació de riscos

	Probabilitat	Impacte
R1	Alta	Catastròfic
R2	Alta	Crític
R3	Baixa	Marginal
R4	Mitja	Marginal
R5	Mitja	Crític
R6	Baixa	Crític
R7	Mitja	Catastròfic
R8	Baixa	Crític
R9	Mitja	Catastròfic
R10	Mitja	Crític
R11	Baixa	Catastròfic

3.2.7. Pla de contingència

	Solució que cal adoptar
R1	Ajornar alguna funcionalitat. Tenir una base ben definida.
R2	Revisar l'estudi de viabilitat, modificar la planificació.
R3	Re negociar amb el club i el director del projecte.
R4	Re negociar amb el club i el director del projecte.
R5	Preveure aquesta situació, millorar el treball de cada membre de l'equip.
R6	Fixar un calendari de reunions, millorar contacte amb el club i el director.
R7	Realitzar tests automàtics i periòdics, donar garanties.
R8	Revisar normes i legislació, consultar amb experts.
R9	Revisar seguretat de cada part de l'aplicació en el moment que s'ha creat.
R10	Augmentar hores de treball dels altres membres de l'equip.
R11	Sense solució.

3.3. Pressupost

3.3.1. Estimació cost de personal

Director projecte (DP)	6h	240,00 €
Responsable projecte (A)	494h	12.350,00 €

Total: 12590 €

3.3.2. Estimació cost de recursos

	Cost amortització	Cost unitari	Període amortització	Període utilització
PC programador	200,00 €	1.200,00 €	36m	6 mesos
Ms office	41,60 €	250,00 €	36m	6 mesos
Ms Project	60,00 €	360,00 €	36m	6 mesos

Total: 301,6 €

3.3.3. Resum i anàlisi cost benefici

Cost de desenvolupament del projecte	12.590,00 €
Cost d'amortització del material	301,60 €
TOTAL	12.891,60 €

El pressupost del projecte s'eleva a una quantitat alta de diners. Hem de tenir en compte les repercussions econòmiques que suposarà l' implementació de l'aplicació. Amb ella reduïrem el consum de paper, reduïrem els diners gastats mensualment en trucades telefòniques entre diferents membres i departaments del club i la reducció d'hores en tasques administratives.

3.4. Conclusions

Beneficis:

- ✚ Millora del control de la gestió econòmica del club.
- ✚ Millora la forma de funcionar del club.
- ✚ Millora del àmbit administratiu i esportiu del club.
- ✚ Reducció de costos en material administratiu.
- ✚ Realització de pressupostos de manera intuïtiva.
- ✚ Estabilitat financera.
- ✚ Millora de la seguretat de la informació.
- ✚ Accés ràpid a registres.
- ✚ Reducció hores de treball administratiu centralitzat.

Inconvenients:

- ✚ Inversió important per l'entitat.
- ✚ Necessitat d'un procés de formació per cadascun dels perfils.

4. El projecte Intranet pel HCR Cent Patins

Intranet pel HCR Cent Patins és el nom que rep el projecte d'evolució administrativa i social per el Club de Hockey Línia Cent Patins.

Aquest projecte neix a finals de l'any 2009 amb l' intenció de millorar les condicions actuals del club i contribuir al seu creixement i expansió. El programari utilitzat fins ara no compleix els requisits funcionals per dur a terme una comptabilitat transparent i segura. Per aquest motiu l'equip del projecte ha apostat per donar un canvi tecnològic al Club amb eines més actuals.

L'equip del projecte ha pensat en una intranet per a poder gestionar les transaccions econòmiques del Club.

S'han utilitzat eines tecnològiques actuals amb llenguatges de programació molt potents. Totes les aplicacions utilitzades al projecte són gratuïtes i de lliure distribució, fet que ha abaratit molt els costos de producció.






4.1. Llenguatges informàtics

Un llenguatge informàtic és un llenguatge utilitzat per, o associat amb, ordinadors.

Dins dels llenguatges informàtics trobem subconjunts com: llenguatges de programació, llenguatges d'especificació, llenguatges de consulta, llenguatges de scripting, llenguatges de marques, llenguatge de transformació, llenguatges de comunicació, llenguatges de so, llenguatges gràfics, llenguatges d'estil i pseudocodi. No hem de confondre aquest terme amb el de llenguatges de programació.

Un llenguatge de programació és un idioma artificial dissenyat per expressar computacions que poden ser dutes a terme per màquines.

Per el desenvolupament de la nostra aplicació hem utilitzat:

-  Llenguatge de programació: PHP.
-  Llenguatge de scripting: JavaScript
-  Llenguatge de marques: HTML
-  Llenguatge de consulta: SQL
-  Llenguatge d'estil: CSS

4.1.1. PHP



És un llenguatge de programació interpretat, dissenyat originalment per la creació de pàgines web dinàmiques¹. És utilitzat principalment en l'interpretació del costat del servidor (server-side-scripting²) encara que actualment pot ser utilitzat des de una interfície de línia de comandes o en la creació d'altres tipus de programes incloent aplicacions amb interfície gràfica.

PHP és un acrònim recursiu³ que significa **PHP Hypertext Pre-Processor**. Va ser creat originalment per Rasmus Lerdorf⁴ al 1994, no obstant la primera implementació important va ser produïda per *The PHP Group*.

La seva llicència és *PHP License 3.0.1*, la *Free Software Foundation*⁵ considera aquesta llicència com a software lliure.

4.1.1.1. Funcionament

Al ser un llenguatge server-side-scripting els codis PHP s'executen al servidor, per això ens permet accedir als recursos que tingui el servidor, com per exemple una base de dades.

El resultat d'aquesta execució s'envia al navegador de l'usuari que està fent la petició, aquest sol ser una pàgina HTML.



¹ Lloc web que permet interactuar amb el visitant, de manera que cada usuari que visita la pàgina vegi la informació modificada.

² La programació del costat del servidor és una tecnologia que consisteix en el processament d'una petició d'un usuari mitjançant l'interpretació d'un script en el servidor web per generar pàgines HTML dinàmicament com a resposta.








³ Un acrònim és recursiu si la primera lletra del mateix fa referència al propi acrònim.

⁴ Programador informàtic nascut a Grenlàndia al 1968.

⁵ La Free Software Foundation és una organització creada al 1985 per el programador americà Richard Stallman. El seu objectiu és eliminar les restriccions sobre la copia, redistribució, enteniment i modificació de programes de computadores.

4.1.1.2. Característiques

Al ser un llenguatge lliure disposa d'una gran quantitat de característiques que el converteixen en l'eina ideal per la creació de pàgines web dinàmiques.




-  Suport per la majoria de bases de dades: MySQL, PostgreSQL, Oracle, MS SQL Server..
-  Integració amb diferents biblioteques externes.
-  Solucions simples i universals per la paginació dinàmica.
-  Manteniment de codi fàcil i senzill.
-  Gran suport on-line per a qualsevol tipus d'error.
-  Continues millores i extensions per ampliar les capacitats del llenguatge.
-  Ofereix seguretat per defecte al ser sempre executat a la banda del servidor.

4.1.2. HTML

Llenguatge de Marcat d'Hipertext (HTML) és el llenguatge de marques predominant per l'elaboració de pàgines web. Es utilitza per descriure l'estructura i el contingut en forma de text, així com per complementar el text amb objectes tals com imatges.

HTML s'escriu en forma d'etiquetes rodejades per cotxets angulars (< >).

4.1.2.1. Característiques

-  Reconegut a tot el món. Les seves normes les defineix l'organisme World Wide Web Consortium, més conegut com W3C⁶.
-  Una pàgina HTML es visualitza de manera similar a qualsevol navegador de qualsevol sistema operatiu.
-  Estructura les pàgines web.

4.1.3. CSS

També conegut com fulles d'estil en cascada, és un llenguatge d'estil utilitzat per definir la presentació d'un document estructurat en HTML o XML⁷. El W3C és l'encarregat de formular l'especificació de les fulles d'estil que serviran d'estàndards per als navegadors.

⁶ El W3C és un consorci internacional que produeix recomanacions per la World Wide Web. El seu objectiu es guiar als programadors a través de protocols i pautes que assegurin el creixement futur de la Web.

⁷ El llenguatge de marques extensible és un metallenguatge extensible d'etiquetes similar al HTML.

Per exemple, si posem un element HTML a la nostra pàgina del tipus encapçalat `<h1>` li podem donar estil amb sentències CSS del tipus:

```
h1 {
    padding: 5 px;
    margin: 10 px;
    color: #FF0000;
}
```

El nostre títol tindrà una aparença diferent a la per defecte de HTML.

4.1.4. JavaScript

És un llenguatge de scripting u orientat a document que ens permet interactuar objectes procedents d'altres aplicacions. No és possible realitzar una aplicació amb JavaScript, però si millorar-ne alguna d'existent.

JavaScript ha tingut influència de múltiples llenguatges i es va dissenyar amb una sintaxis semblant al llenguatge de programació Java.

Aquest llenguatge va ser inventat per Brendan Eich⁸ de l'empresa Netscape Communications. El seu llançament va ser anunciat entre Sun Microsystems i Netscape al 1995.

Serveix principalment per millorar la gestió de d'interfície client/servidor. Un script JavaScript inserit en un document HTML permet reconèixer i tractar localment (en el client) els esdeveniments generats per l'usuari, en ocasions poden gestionar el recorregut del document HTML o formularis.

4.1.4.1. Característiques

- 🚦 És simple i eficaç.
- 🚦 Manipula objectes dintre de la nostre pàgina web. Podem definir esdeveniments a qualsevol dels objectes.
- 🚦 Molt utilitzat per al control i validació de formularis. Permet tenir estructures més segures i robustes.
- 🚦 És dinàmic, respon als esdeveniments en temps real.
- 🚦 Permet l'ús d'expressions regulars⁹.

⁸ Programador americà nascut al 1961. Involucrat en projectes de Mozilla Corporation.

⁹ Conjunt de mètodes que permeten la validació de formats específics en una cadena de caràcters donada.

4.1.4.2. Biblioteca JQuery

En el nostre projecte també ens hem interessat per la biblioteca JQuery de JavaScript.

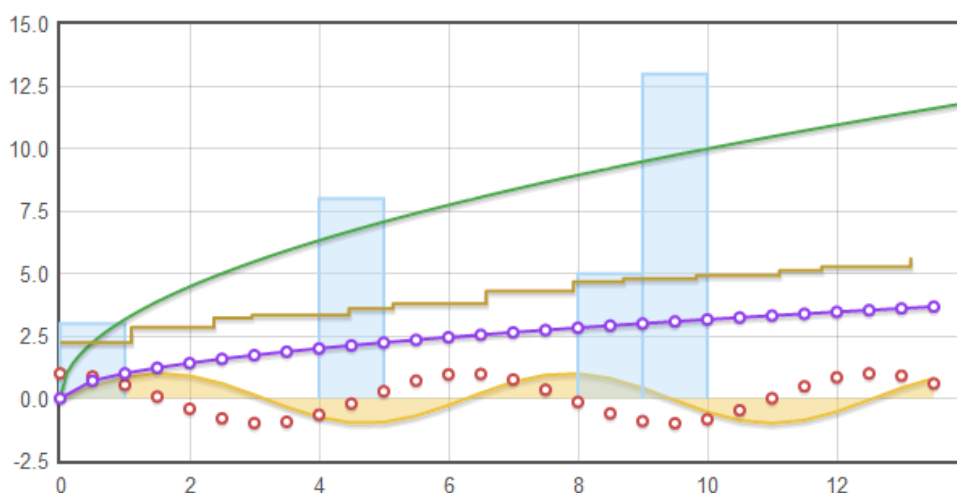
JQuery ens permet simplificar la manera d'interactuar amb els documents HTML, controlar esdeveniments, desenvolupar animacions i agregar interacció amb la tecnologia AJAX¹⁰ a pàgines web.

JQuery és software lliure i de codi obert amb una llicència de GNU.

Nosaltres ens hem interessat en aquestes llibreries per a poder inserir gràfics de control econòmic a la nostra aplicació i elements que permeten interactivitat amb els usuaris.

4.1.4.2.1. Característiques

- ✚ Selecció d'elements DOM¹¹.
- ✚ Interactivitat i modificacions de l'arbre DOM.
- ✚ Esdeveniments.
- ✚ Manipulació de la fulla d'estils CSS.
- ✚ Efectes i animacions.
- ✚ AJAX
- ✚ Suporta extensions.



¹⁰ És una tècnica de desenvolupament web per crear aplicacions interactives (RIA). Aquest tipus d'aplicacions s'executen en el client, és a dir, mantenen una comunicació asíncrona amb el servidor.

¹¹ El model en objectes per la representació de documents (DOM) es una interfície de programació d'aplicacions que proporciona un conjunt estàndard d'objectes per representar documents HTML i XML.

4.1.5. SQL

El llenguatge de consulta estructurat (SQL) és un llenguatge declaratiu d'accés a bases de dades relacionals que permet especificar diversos tipus d'operacions en elles.

Una de les característiques és l' utilització d'àlgebra i càlcul relacional per efectuar consultes amb l'objectiu de recuperar informació d'interès d'una base de dades.

Aquest llenguatge és de quarta generació (4GL¹²), d'alt nivell i de no procediment gràcies a la seva forta base teòrica i orientació a la manipulació de conjunt de registres, i no registres individuals. Permet una alta productivitat en codificació i orientació d'objectes.

El llenguatge SQL està dividit segons les funcionalitats que volem estendre:

4.1.5.1. DDL

El llenguatge de definició de dades (DDL) és el que s'encarrega de la modificació de l'estructura dels objectes de la base de dades, existeixen quatre operacions bàsiques: CREATE, ALTER, DROP i TRUNCATE.

4.1.5.2. DML

El llenguatge de manipulació de dades (DML) és un llenguatge proporcionat per el sistema de gestió de base de dades que permet als usuaris dur a terme tasques de consulta o manipulació de les dades, organitzades per el model de dades adequat. Hi ha operacions tres bàsiques: INSERT, UPDATE, DELETE.

4.2. Eines tecnològiques

Ens referim a eines tecnològiques com el conjunt de programes i aplicacions (software) que són dissenyades per a facilitar el treball i permetre que els recursos siguin aplicats eficientment intercanviant informació i coneixement dins i fora de les organitzacions que les subministren.

Per al desenvolupament del nostre projecte hem utilitzat diferents eines que ens han ajudat a crear diferents aspectes de la nostra aplicació.

Hem utilitzat un framework que ens ha estalviat moltes línies de codi, un servidor on allotjar la nostra feina i un dissenyador d'estructures del tipus base de dades.

¹² Tipus de llenguatges de programació en els que s'especifica quins resultats són els que es volen obtenir, a diferencia dels de 3GL, on s'especifica com s'han d'obtenir.

4.2.1. Joomla 1.5



Joomla! és un sistema de gestió de continguts (CMS). entre les seves principals virtuts destaquen la de permetre editar el contingut d'un lloc web de manera molt intuïtiva.

És una aplicació de codi obert majoritàriament en PHP sota una llicència GPL¹³.

Aquest gestor de continguts pot treballar a Internet o a intranets i requereix una base de dades MySQL així com un servidor HTTP Apache.

A Joomla! s'inclouen característiques com: millorar el rendiment web, versions imprimibles de pàgines, flash amb notícies, blocs, fòrums, enquestes, calendaris, cerques den el lloc web..

El seu nom és una pronunciació fonètica per anglòfons de la paraula swahili *jumla*, que significa "tots junts" o "com un tot". Aquest nom es va escollir com una reflexió de compromís del grup de desenvolupadors i la comunitat del projecte.

On sorgeix aquest projecte?

Joomla sorgeix com a resultat d'una bifurcació o millora de Mambo¹⁴, de la corporació Miro d'Austràlia. El grup de programadors de l'empresa decideix emprendre un nou camí i desenvolupar Joomla!, un projecte obert i sense ànim de lucre.

El grup de desenvolupadors va crear un lloc web denominat OpenSourceMatters per distribuir informació als usuaris, desenvolupadors, dissenyadors web i a la comunitat en general.

Tothom pot formar part d'aquest projecte i desenvolupar aplicacions sota la llicència GNU. Desenvolupadors d'arreu del món creen i distribueixen els seus productes.

¹³ La Llicència Pública General o coneguda com GNU, és una llicència creada per la Free Software Foundation (FSF) a 1980, i està orientada principalment a protegir la llibre distribució, modificació i ús de software.

¹⁴ Sistema de portals CMS basat en el llenguatge de programació PHP i base de dades SQL de codi obert. La llicència d'aquest gestor de continguts es de la empresa australiana *Miró*.

Com s'estructura aquest projecte si és un projecte global?

La gestió de l'administració principal del projecte esta delegada al grup principal ("Core Team"). Tots els membres d'aquest grup treballen en conjunt com un sòl grup, compromesos a guiar a Joomla! dins del moviment de codi obert. Aquest grup està compost per diferents perfils amb variada experiència i una sèrie diversa de disciplines.

El grup neix al 2005 juntament amb el projecte. És molt més que una congregació de desenvolupadors, la seva responsabilitat principal radica en l'organització de Joomla! com a aplicació (estructura funcional) i l'estructura de la comunitat.



El grup principal ("Core Team") és l'encarregat de coordinar i gestionar els diferents departaments o comunitats. Aquestes comunitats són obertes, per tant qualsevol persona pot col·laborar en el projecte en algun dels seus àmbits.

Si nosaltres desitgéssim contribuir a la comunitat amb el nostre projecte, podríem publicar el nostre codi font a l'apartat de *Developers* de la Comunitat Joomla!¹⁵. El procés de desenvolupament, producció i llançament dins de la Comunitat segueix uns patrons per tal de tenir un control de versions i assegurar el bon funcionament de totes les aplicacions.

¹⁵ <http://developer.joomla.org/>



A la fase *Alpha* es fixen uns objectius, es desenvolupen, s'integren al sistema (Framework) i es creen les extensions (instal·lables). La primera part d'aquest procés correspon a la fase d'Anàlisi (*Set Goals*) posteriorment es refactoritzen les tasques i es comença a desenvolupar (*Refactor & Develop*). La fase *Beta* és on es creen els Tests unitaris i es crea la documentació necessària per al llançament de l'aplicació.

Per últim, s'estabilitza l'aplicació (fins ara una Beta¹⁶) i es crea la primera versió del producte. Es llança a la comunitat per poder ser distribuït arreu de la xarxa. Per poder entendre la complexitat d'aquest projecte i els cicles de creació i llançament dels productes cal que entenguem les seves característiques tècniques:

4.2.1.1. El CMS

El Gestor de Continguts de Joomla! ens permet focalitzar els nostres esforços en la informació, deixant a banda la part més costosa, la gestió de dades a un lloc web.

El CMS de Joomla! ens ofereix:

- Organització del lloc Web: Notícies organitzades en seccions i categories. Aquesta arquitectura ens permet tenir la informació ordenada i de fàcil accés i manipulació.

¹⁶ Una versió beta representa generalment la primera versió completa d'un programa informàtic que es possible que sigui inestable però útil per a les demostracions.

- Publicació de continguts: Podem crear, editar i eliminar pàgines amb un simple editor que permet donar format a texts amb estils o imatges.
- Escalabilitat e implementació de noves funcionalitats: Joomla! ens permet la possibilitat d'instal·lar, desinstal·lar i administrar nous components i mòduls. Més endavant en parlarem més detalladament.
- Administració d'usuaris: Funcionalitat que ens permet emmagatzemar dades d'usuaris registrats i també la possibilitat d'enviar e-mails massius a tots els usuaris o grups d'usuaris. Hi ha diferents grups d'usuaris dividits per jerarquies segons el nivell de permisos dins de la gestió i administració del lloc.
- Disseny i aspecte estètic del lloc: Es possible canviar tot l'aspecte mitjançant un sistema de *templates*¹⁷.
- Navegació i menú.
- Administrador d'imatges: Emmagatzemades al servidor.
- Disposició de mòduls modificable: La posició dels mòduls pot col·locar-se com es desitgi
- Enquestes.
- Publicitat: Es possible fer publicitat utilitzant l'Administrador de Baners.
- Estadístiques de visites: Amb informació del tipus de navegador, S.O i detalls de documents més vistos.


4.2.1.2. Arquitectura de Joomla

El CMS de Joomla! és molt potent i ens ofereix moltes funcionalitats que encaixen perfectament amb alguns dels requisits funcionals del nostre projecte, com per exemple la gestió d'usuaris.


La resta del nostre projecte, la part més específica i personalitzada s'haurà de desenvolupar en un entorn determinat i utilitzant l'estructura que ens ofereix Joomla!.

¹⁷ Plantilla pre-desenvolupada amb un patró, disseny i estil per a pàgines web.


Joomla 1.5 es descompon en tres capes:

 *Joomla Extensions*: Hem comprovat que Joomla! té una extensa llista de característiques i funcionalitats per defecte. No obstant, no tot el que vulguem per al nostre web està disponible. Per aquest motiu hi ha les extensions. Són normalment petites aplicacions, plugins, plantilles o packs de llenguatge que fan possible que Joomla creixi constantment. Hi ha cinc tipus d'extensions: components, mòduls, plugins, templates i llenguatges.

Més endavant aprofundirem en aquests tipus i explicarem en quin d'aquests focalitzarem la nostra aplicació per a la gestió de Club.

 *Joomla Applications*: En aquesta capa de l'arquitectura trobem l'aplicació Joomla!. Aquesta és composta d'un Front-End¹⁸ (classe¹⁹ JSite), un Back-End (classe JAdministrator), un instal·lador (classe JInstallation) que ens permet implantar el sistema a qualsevol servidor dotat amb Apache, PHP i una base de dades. També trobem una classe amb el protocol XML-RPC. Aquesta classe permet instanciar un protocol de trucada a procediment remot utilitzant XML²⁰ per codificar les dades i HTTP²¹ com a canal de transmissió de missatges. Aquestes sub-aplicacions estan entrelaçades, fet que ens permet tenir un sistema robust i segur.

Igual que amb les extensions, Joomla! ens ofereix un entorn lliure per a poder desenvolupar les noves aplicacions en aquesta capa.

 *Joomla Framework*: Conjunt de classes de Joomla que ens permeten desenvolupar aplicacions. Això ens permet la reusabilitat del codi font i avantatges a l'hora de programar les nostres aplicacions. *Joomla Applications* i *Joomla Extensions* estan desenvolupats utilitzant Joomla Framework. En els pròxims punts donarem una visió més específica sobre el Framework.

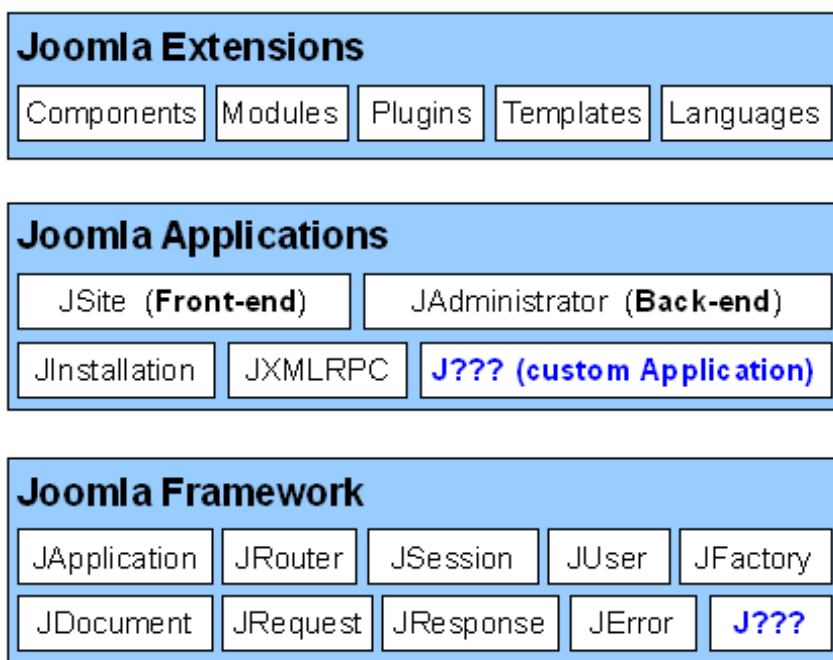
¹⁸ El Front-End és la part del software que interactua amb els usuaris. El Back-End és la part que processa l'entrada desde el Front-End.

¹⁹ En programació orientada a objectes, una classe és una construcció que s'utilitza com a model per crear objectes de la classe. Aquest model descriu l'estat i el comportament que tots els objectes de la classe comparteixen.

²⁰ El Llenguatge de Marques Extensible (XML) és un metallenguatge extensible d'etiquetes desenvolupat per el W3C. Permet definir la gramàtica de llenguatges específics.

²¹ El Protocol de transferència d'Hipertext (HTTP) és el protocol utilitzat en cada transacció de la World Wide Web (WWW). HTTP defineix la sintaxis i la semàntica que utilitzen els elements de software de l'arquitectura web (clients, servidors, proxies) per comunicar-se.

Architecture of Joomla 1.5.x



4.2.1.3. Les capes superiors: Extensions & Applications

Com hem comentat anteriorment, Joomla Extensions ens permet estendre les funcionalitats que el sistema ens ofereix per defecte. En el nostre cas, hem decidit crear una petita aplicació per Joomla! que es correspon a la part de components i mòduls. Es per això que ens limitarem a explicar les parts involucrades en el nostre desenvolupament.

Que és un component?

Denominem components a petites aplicacions independents entre si que gestionen informació dintre del CMS Joomla!. Els components afegeixen noves i diverses funcionalitats a la nostra web.

Alguns components estan relacionats amb les funcionalitats principals del CMS, com per exemple la gestió d'articles o la gestió d'usuaris. D'altres són addicionals i ens permeten funcionalitats extres.

Que és un mòdul?

Els mòduls són petites aplicacions que mostren informació secundària, simplificada i breu procedent normalment dels components. Els mòduls per exemple ens diuen qui està on-line, quins són els últims articles publicats etc.

La diferencia principal entre els components i els mòduls radica en la complexitat de les aplicacions. Normalment els components són aplicacions notablement més grans que els mòduls. S'encarreguen de la lògica de negoci i del tractament de dades amb la base de dades. Els mòduls posteriorment donen un detall de les operacions realitzades per algun component associat. Els components poden operar també dins de la intranet d'administració que ens facilita Joomla! més conegut com a Back-End. Des de aquesta zona del nostre lloc web podem tenir control sobre totes les petites aplicacions que viuen amb el sistema.

Si volem que els resultats obtinguts per els components siguin públics al Front-End cal que inserim un mòdul, especialment destinat per aquest fi, en alguna posició del template de la pàgina que desitgem. Aquest mòdul sempre serà visible per a tots els usuaris.

D'aquesta manera quan un usuari visita una de les nostres pàgines del Front-End s'executa el següent procés:



Podrem afegir tots els mòduls que vulguem a les nostres pàgines.

Alguns dels mòduls més comuns són: menús, galeries de fotografies, enquestes, embolcalls..

Vista d'un component des de el Back-End, en aquest cas el component *Gestor d'enquestes*:



Vista del mòdul d'enquestes, l'encarregat de mostrar als usuaris les nostres enquestes dissenyades al component:

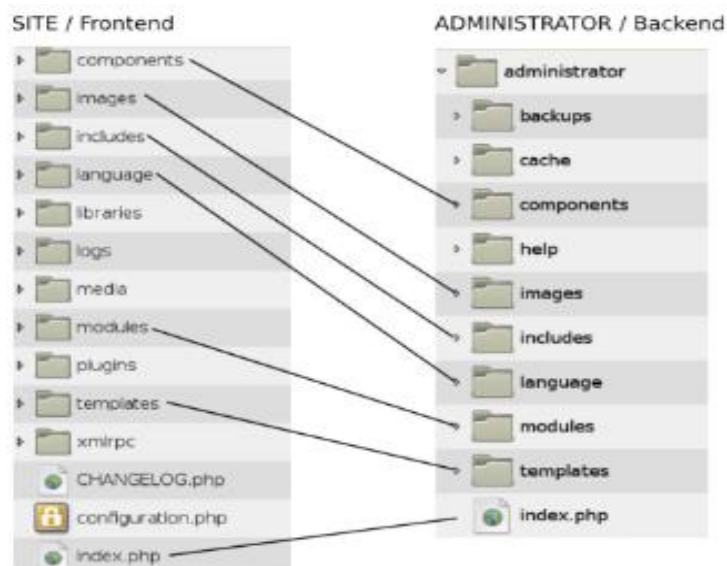
Enquesta

¿Para qué usas Joomla!?

- ☐ Comunidades-Grupos
- ☐ Sitios públicos
- ☐ Comercio electrónico
- ☐ Blogs
- ☐ Intranets
- ☐ Fotos y sitios multimedia
- ☐ Para todo lo anterior!

Existeixen mòduls i components per a totes dues parts, al Back-End i al Front-End.

Independentment de on siguin els mòduls i components, aquests comparteixen llibreries del Framework i mateixa base de dades. De tal manera que tenim dues parts molt diferenciades però que comparteixen recursos.



Observem la similitud entre les dues zones.

En el nostre cas, nosaltres hem optat per desenvolupar un component per el Back-End, això ens proporcionarà un sistema de seguretat integrat a Joomla Applications. L'arquitectura dels components també facilita molt el desenvolupament i ens permet estalviar moltes línies de codi. El patró utilitzat és el Model-Vista-Controlador (MVC).

4.2.1.3.1. Model-View-Controller

És un estil d'arquitectura de software que separa les dades d'una aplicació, d'interfície d'usuari i la lògica de control en tres components diferents. Aquest estil és comú a aplicacions web.

Aquest patró va ser descrit per primer cop al 1979 per Trygve Reenskaug²².

Els diferents components d'aquesta arquitectura són:

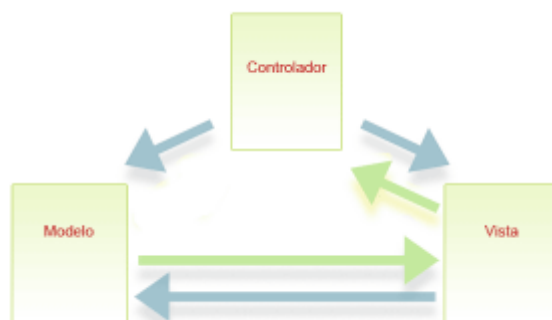
- 🔗 Controlador: Encarregat de controlar l'accés a la nostra aplicació. Invoca peticions al model i a la vista.
- 🔗 Model: Membre del controlador encarregat de manipular les operacions lògiques i de tractament d'informació.

²² Professor d'informàtica de l' Universitat d'Oslo nascut al 1930. Gran investigador de la programació orientada a objectes. Fundador de la companyia d'IT, Taskon.

- ✚ Vista: Encarregat d'expressar l'última forma de les dades; d'interfície gràfica que interactua amb l'usuari final del programa (GUI²³). Evidencia tota l'informació obtinguda en el procés.

Flux del patró MVC:

- ✚ L'usuari interactua amb d'interfície d'usuari. Per exemple, click a un botó.
- ✚ El controlador rep la notificació de l'acció sol·licitada per l'usuari. El controlador gestiona l'esdeveniment que arriba.
- ✚ El controlador accedeix al model, amb una petició relacionada amb l'acció sol·licitada per l'usuari.
- ✚ El controlador delega a les objectes de la vista la tasca de desplegar d'interfície d'usuari. La vista obté les dades del model per generar l' interfície apropiada per a l'usuari on es reflexen els canvis en el model.
- ✚ L' interfície d'usuari espera noves interaccions de l'usuari, començant el cicle novament.



Joomla! ens permet aquest tipus d'arquitectura ja que l'entorn és orientat a objectes.

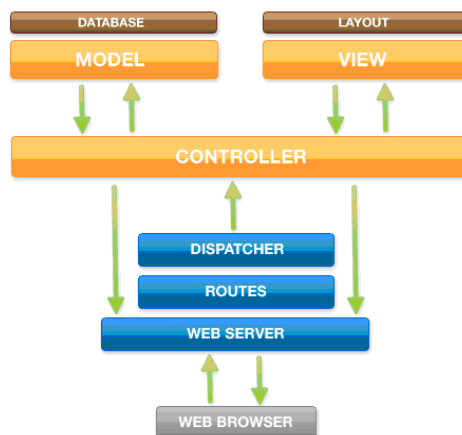
El controlador serà invocat per la classe JDispatcher, classe que hereta de JEvent. La base del llançament del controlador serà per esdeveniments.

Quan volem obtenir dades del model, Joomla! es connectarà a la base de dades MySQL per fer les comandes necessàries prèviament definides al model.

²³ L'Interfície Gràfica d'Usuari (GUI) és l'artefacte tecnològic d'un sistema interactiu que possibilita una interacció amigable amb un sistema informàtic.

Per mostrar les interfícies d'usuari el sistema utilitza layouts dissenyats en HTML i CSS.

JOOMLA MODEL VIEW CONTROLLER (MVC)



4.2.1.4. Les entranyes de Joomla: el Framework

Baixem una capa més i arribem a les entranyes del sistema Joomla!, el framework.

Que és un framework?

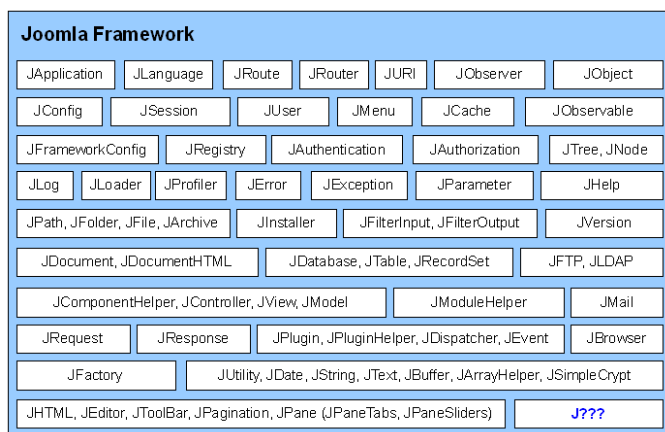
Un *framework* és una estructura conceptual i tecnològica de suport definit, normalment amb mòduls de software concret que permet que altres projectes de software puguin ser organitzats i desenvolupats. Normalment inclouen programes, biblioteques i un llenguatge interpretat per ajudar a desenvolupar i unir diferents components d'un projecte.

Dit en altres paraules, un framework és un disseny re usable per algun sistema de software.

Diferents parts d'un framework poden ser exposades mitjançant una *application programming interface* (API²⁴).

²⁴ Una interfície de programació d'aplicacions (API) és el conjunt de funcions i procediments (mètodes en programació orientada a objectes) que ofereix una biblioteca per ser utilitzats per un altre software com una capa d'abstracció.

Com és Joomla Framework?



Si ens fixem en l'imatge anterior observem un conjunt de noms que casualment comencen tots per "J". Tots aquests termes fan referència a classes genèriques de Joomla!.

Podem dir que el conjunt d'aquestes classes forma una estructura conceptual i tecnològica que ens permet desenvolupar altres solucions com components o mòduls, és a dir, un framework. Totes les classes que tenim a Joomla! les podem utilitzar a l'hora de desenvolupar la nostra aplicació.

De la mateixa manera que Joomla Extensions i Joomla Applications ens permeten la creació i personalització del sistema, el Framework també ens brinda l'oportunitat de modificar les classes existents o crear-ne de noves. Totes les capes del sistema són completament Open Source²⁵, així que no és estrany trobar diferents versions del Framework modificats per programadors d'arreu del món.

Per aprofundir una mica més, podem observar algunes de les classes de Joomla Framework que han sigut utilitzades en aquest projecte.

4.2.1.4.1. JFactory

Aquesta classe ens permet accedir a diferents objectes de caràcter global necessaris pel Framework.

Els mètodes de la classe ens retornen objectes com la data (getDate), la base de dades utilitzada a l'aplicació des de la que es fa la crida (getDBO) o el llenguatge utilitzat en aquell moment (getLanguage), d'entre altres.

²⁵ Terme amb el que es coneix el software distribuït i desenvolupat lliurement. L'Open Source té un punt de vista més orientat als beneficis pràctics de compartir el codi que a les qüestions morals i/o filosòfiques, de les quals destaquen el moviment de GNU.

4.2.1.4.2. JRequest

Proveeix una interfície per accedir a variables per petició. Inclou el pas de variables per "\$_POST", "\$_GET" i "\$_REQUEST"²⁶.

4.2.1.4.3. JLoader

Permet importar i carregar parts de les nostres aplicacions. Utilitzant els mètodes d'aquesta classe podem carregar una aplicació per parts. Això permet agilitat ja que només carregarà les classes que intervenen en l'inici de l'aplicació.

4.2.1.4.4. JToolBarHelper

Classe d'ajuda per simplificar la creació de títols, botons i tota mena d'accessoris d'una Toolbar²⁷. És utilitzada en el Back-End del sistema.

4.2.1.4.5. JHTML

Classe que ens permet inserir tot tipus de controls HTML a les vistes de la nostra aplicació. Podem inserir des de elements bàsics, com botons o links, fins a elements més sofisticats com calendaris, imatges o scripts.

4.2.1.4.6. JController

Joomla! ens ofereix la possibilitat de treballar amb una arquitectura de tres capes. Podem crear instàncies que heretin de JController per crear controladors.

Més endavant explicarem els models de tres capes.

Els mètodes d'aquesta classe ens possibiliten la renderització de vistes, redireccionament, assignar tasques a botons de la vista, etc.

4.2.1.4.7. JModel

Un dels altres nivells de l'arquitectura de tres capes.

Aquesta classe ens proporciona mètodes per connectar-nos a la base de dades i treballar amb ella. Això ens estalvia moltes redundàncies i sovint moltes practiques de hardcoding²⁸.

²⁶ Mètodes d'enviament i recepció de dades procedents de formularis nadius del llenguatge de programació PHP.

²⁷ En català, barra d'eines. Component d'una interfície gràfica d'usuari (GUI) mostrada usualment per la pantalla com una fila, columna o bloc que conté icones o botons que ens permeten activar diferents funcionalitats d'una aplicació.

4.2.1.4.8. JView

Nivell superior de l'arquitectura de tres capes.

Conté tots els mètodes involucrats en la presentació de les dades per pantalla, carga de templates, assignació de layouts, etc.

4.2.1.5. Seguretat

La seguretat i el correcte tractament d'errors són imperatius per a qualsevol sistema computacional. Sistemes que són a la World Wide Web amb seguretat pobre tenen un alt risc.

Joomla! ens ofereix la seguretat, fiabilitat i tractament d'errors del sistema que desitgem per a una aplicació.

4.2.1.5.1. Tractament d'errors

Quan trobem errors és important que es prenguin en consideració. Per aquest motiu a Joomla! hi han les classes JError i JException que faciliten molt les tasques. Amb les classes podem classificar errors i personalitzar les accions de resposta com desitgem.

Hi ha les següents accions: Ignorar, mostrar missatge, mostrar missatge amb la traça d'error, matar un procés, afegir un missatge a la cua de missatges, triggers i crides a altres mètodes.

4.2.1.5.2. Prevenció SQL Injection

La injecció SQL és una vulnerabilitat informàtica en el nivell de validació de les entrades a la base de dades d'una aplicació. L'origen es el filtrat incorrecte de les variables utilitzades en les parts del programa amb codi SQL. Joomla! al ser un sistema que treballa amb bases de dades amb llenguatge SQL està exposat a aquests atacs.

Tenim diferents mètodes al Framework per detectar aquests atacs. Un d'ells prové de la classe JRequest, utilitzada per el pas de variables entre la vista i el controlador. El seu funcionament es basa en la comprovació de les dades enviades posteriorment d'haver indicat el seu tipus. Si indiquem que una variable serà del tipus numèric no recollirà cap valor alfabètic.

²⁸ Fa referència a la mala practica en el desenvolupament de software. Consisteix en incrustar dades directament en el codi font del programa en comptes d'obtenir les dades d'una font externa com un fitxer de configuració o un arxiu de recursos.

Un altre mètode és utilitzar prefixos per a caràcters especials utilitzant una barra "\". Joomla! utilitza els mètodes Quote() i getEscaped() de la classe JFactory per encapsular les dades quan fem les consultes.

4.2.1.5.3. Expressions Regulars

Les expressions regulars s'utilitzen també per a la prevenció d'atacs per injecció, encara que el seu principal objectiu és comprovar el format de les dades i modificar-les en cas de que siguin errònies. Hi ha molts patrons utilitzats per a la validació de dades, delimitadors, quantificadors, etc.

4.2.1.5.4. Control d'accés

Joomla! utilitza mecanismes de control d'accés segons grups d'usuaris GACL (Group Acces Control Lists). Dins d'aquest grups hi ha subgrups amb diferents permisos. Un cop es fa l'autenticació, el sistema emmagatzema les variables de sessió. Les variables tenen un temps determinat, si es detecta inactivitat durant cert període s'eliminen.

Aquests mecanismes són extensibles al panell de control del Back-End i a les extensions i components privats.

4.2.2. MySQL



Per a desenvolupar la nostra aplicació sobre l'entorn de Joomla! calia que ens adaptéssim a les necessitats del sistema. Com hem explicat anteriorment, a Joomla! les dades són emmagatzemades en bases de dades MySQL.

MySQL és un sistema de gestió de base de dades relacional, multi-fil i multi-usuari amb més de sis milions d'instal·lacions. Aquest sistema de gestió és desenvolupat per Sun Microsystems sota una llicència GNU. Encara que la llicència general sigui d'aquest tipus, Sun Microsystems és una empresa privada i no tot el codi del sistema és obert. La major part del codi té copyright²⁹. És a dir, el sistema pot ser utilitzat per qualsevol persona de manera gratuïta, però, no en podem saber el contingut.

²⁹ En Català, dret d'autor. Conjunt de normes i principis que regulen els drets morals i patrimonials que la llei concedeix als autors pel fet de crear una obra literària, artística, científica o didàctica sigui publicada o inèdita.

L'objectiu de MySQL és complir l'estàndard SQL sense sacrificar velocitat, fiabilitat o usabilitat.

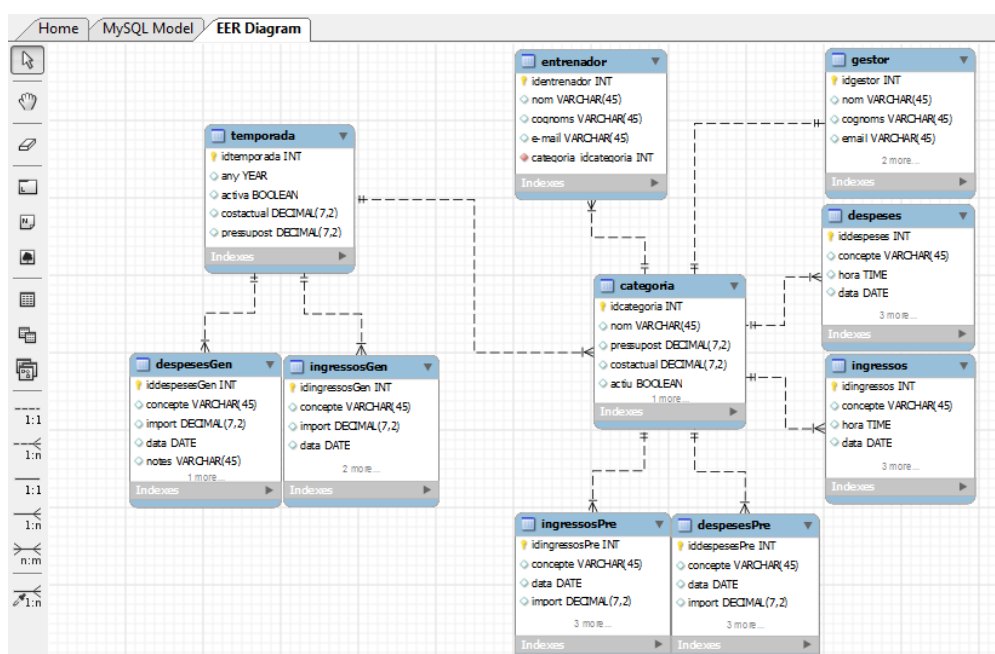
4.2.2.1. MySQL Workbench 5.2

Per a dissenyar les estructures necessàries per desenvolupar la nostra aplicació hem utilitzat el software MySQL Workbench 5.2.

Aquesta eina de Sun Microsystems està destinada a modelar diagrames d'entitat-relació per bases de dades MySQL.

L'aplicació elabora una representació visual de les taules, vistes, procediments emmagatzemats i claus foranies de la base de dades. A més, es capaç de sincronitzar el model de desenvolupament amb la base de dades real.

En el nostre cas després de dissenyar l'arquitectura de les taules, l'aplicació ens ha generat un guió per crear la base de dades.



4.2.3. XAMPP



XAMPP és un servidor independent de plataforma, software lliure, que consisteix principalment en la base de dades MySQL, el servidor Web Apache i els intèrprets per llenguatges de script: PHP i Perl.

El nom prové de l'acrònim **X** (per qualsevol dels diferents sistemes operatius), **A**pache, **M**ySQL, **P**HP i **P**erl.

El programa es distribueix sota la llicència GNU i actua com un servidor Web lliure, fàcil d'utilitzar i capaç d'interpretar pàgines dinàmiques.

Aquesta eina ens permet testear la nostra feina a les nostres pròpies màquines sense tenir que allotjar les nostres aplicacions a un servidor real. XAMPP també permet la funcionalitat de servidor de llocs Web real, encara que normalment és utilitzat com a servidor local.

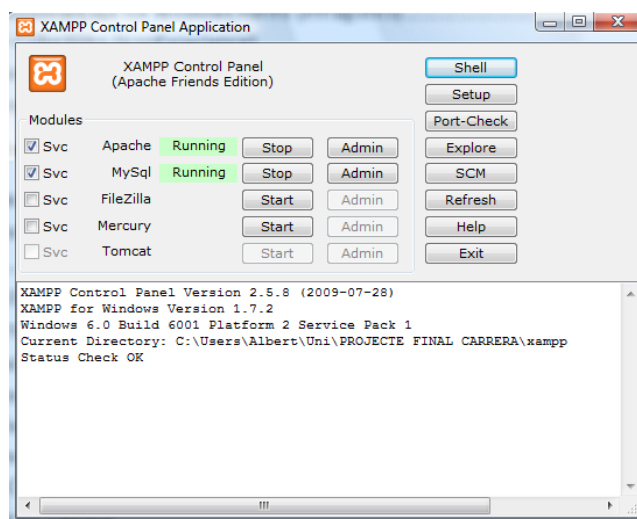
4.2.3.1. Instal·lació de Joomla 1.5 a XAMPP

XAMPP posseeix tots els requisits per a que Joomla! pugui ser executat amb èxit: una base de dades MySQL, un servidor Apache i PHP superior a la versió 4.0.

La instal·lació de Joomla! a XAMPP és equivalent a instal·lar-ho en un servidor real.

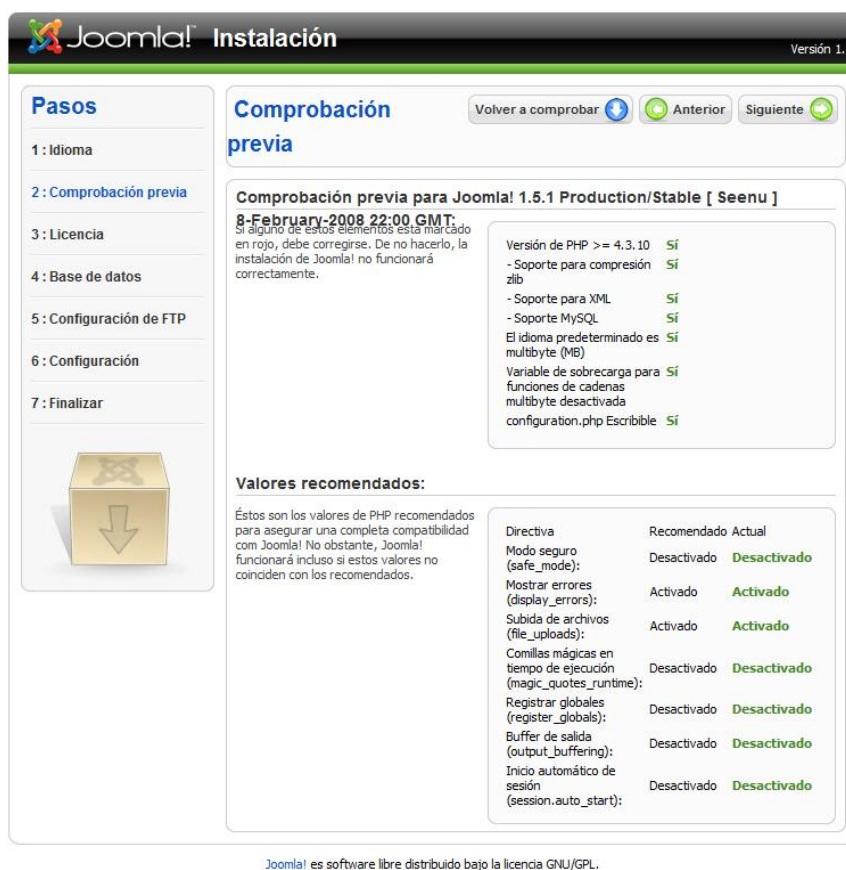
Els passos són els següents:

- 🔗 Instal·lar XAMPP a la nostra màquina. Podem descarregar el fitxer des de la web ApacheFriends.org i executar l'arxiu instal·lable.
- 🔗 Un cop instal·lat amb èxit el servidor local, l'executem.



Ens apareix un panell de control com el de la imatge anterior. Aquí podem iniciar el servidor Apache, la base de dades MySQL i altres funcionalitats que ens ofereix XAMPP.

- ✚ Tot seguit descarreguem Joomla! des de la seva pàgina web Joomla.org. Obtenim un fitxer comprimit.
 - ✚ Descomprimim el fitxer a la carpeta *xampp/htdocs/CarpetaNom*. Aquesta ruta estarà ubicada on hem indicat anteriorment a la instal·lació de XAMPP. CarpetaNom ha de ser el nom que desitgem posar.
 - ✚ Obrim l'explorador Web (és indiferent si és IE, Chrome, Safari..) i a la barra de direccions escrivim:
<http://localhost/CarpetaNom>
- Trobem la següent pantalla:



Observem els passos per instal·lar l'aplicació a la barra esquerra.

- ✚ Per últim, seguir tots els passos fins a que ens doni un missatge de procés completat.

Un cop acabada la instal·lació podrem accedir a la nostra nova web amb entorn Joomla! a la mateixa direcció web que hem utilitzat per començar el procés d'instal·lació.

4.3. La metodologia del projecte

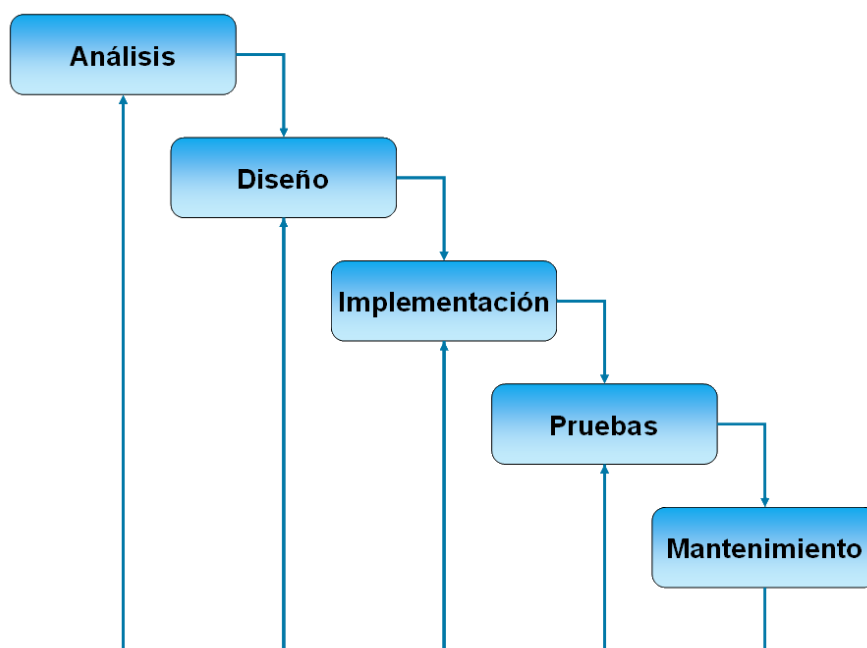
S'ha seguit una metodologia concreta, on s'hi involucren en cada etapa diferents perfils.

Podem dir que hem utilitzat una metodologia similar al desenvolupament en cascada, una tècnica molt estesa a l'enginyeria del software.

4.3.1. Desenvolupament en cascada

També conegut com model en cascada, és una metodologia molt estesa en projectes tecnològics. Les seves principals característiques són l'organització i la sincronització de les seves fases. L'objectiu d'aquest model és planificar, tenir els temps d'execució de les fases ben definits. Sovint és utilitzat en el món empresarial degut a la pressió que tenen les empreses per entregar els seus productes a temps. L'estudi dels recursos i de la productivitat és essencial per poder plantejar aquesta metodologia amb èxit.

Podem distingir les següents fases:



En el nostre cas nosaltres ens trobem en la fase de proves ja que la nostra aplicació es troba en fase Beta, fase prèvia a una versió.

A continuació definirem aquestes etapes i enumerarem les tasques que s'han desenvolupat en cada una d'elles.

4.3.1.1. Anàlisi

Separem dues parts importants l'anàlisi del sistema i dels requisits de software.

A l'anàlisi del sistema s'escull la plataforma en la que es treballarà i es defineixen les tecnologies necessàries en les que es desenvoluparà l'aplicació. S'han de definir tots els elements del sistema que intervenen en el producte final, ja sigui pel seu desenvolupament o per el seu llançament real.

En l'anàlisi dels requisits del software es defineix el problema a resoldre i s'analitzen les necessitats dels usuaris finals de software per tal de conèixer els objectius i especificar-ne tots els requisits de l'aplicació. Aquesta fase es plasma en un document anomenat Document d'Especificació de Requisits (SRD).

Els requisits descriuen el comportament d'un sistema que es vol desenvolupar. Poden ser dividits en casos d'ús (requisits funcionals) que descriuen totes les interaccions que tindran amb els usuaris finals del software o altres característiques que no afecten en les funcionalitats de l'aplicació però intervenen en aspectes de disseny o restriccions (requisits no funcionals).

En aquesta primera part del cicle de vida del software cal tenir una concepció real dels recursos disponibles des del moment en que s'inicia el cicle fins al previst final d'aquest. Els recursos poden tenir perfils molt diferents. Poden ser analistes, dissenyadors, programadors o testers.

És de vital importància que l'analista responsable d'aquesta etapa del projecte tingui molta comunicació i enteniment amb el client. Una mala interpretació de la problemàtica o dels requisits pot desembocar en una desviació del projecte en alguna de les següents fases amb conseqüències fatals tals com pèrdues econòmiques.

Al nostre projecte en aquest pas vam identificar quin era el problema del client: la gestió econòmica del Club.


Després de conèixer el problema concret vam aprofundir en petits requisits tan funcionals com no funcionals que poguessin satisfer les necessitats. El document referent a aquesta fase es troba al punt 2 de la memòria, l'Estudi de Viabilitat.


4.3.1.2. Disseny del Sistema

Es planteja una solució integral d'alt nivell per al desenvolupament del producte. Es descompon i s'organitza el sistema en elements que puguin elaborar-se per separat, aprofitant les avantatges del desenvolupament per equip. Com a resultat sorgeix el Document de Disseny de Software (SDD).

L'objectiu d'aquesta etapa és definir l'estructura de la solució identificant ordres de treball (conjunt de funcions que estaran associades) i les seves relacions. Un cop s'han identificat les ordres de treball es dissenyen a alt nivell les subestructures per a cada una d'elles. El conjunt d'aquestes estructures formarà l'estructura global de la solució.

Aquests procediments són formalment coneguts com estructura de dades i arquitectura de software.

-  Estructura de dades: Forma d'organitzar conjunts de dades elementals amb l'objectiu de facilitar la seva manipulació. S'han d'identificar el conjunt de dades necessaris per a la nostra aplicació i formar una estructura que permeti l'alta, la baixa i la modificació d'aquestes.

-  Arquitectura de software: És el disseny de més alt nivell de l'estructura d'un sistema. Consisteix en un conjunt de patrons i abstraccions coherents que proporcionen un marc de referència necessari per guiar la construcció del software per un sistema informàtic. Aquesta arquitectura es selecciona i es dissenya en base a uns objectius i restriccions definits a la fase d'anàlisi. Aquests no han de ser únicament del tipus funcional, responen a altres objectius com la mantenibilitat, flexibilitat e interacció amb altres sistemes d'informació. Per tant es defineixen els components, les seves interfícies i la comunicació entre elles.

Nosaltres vam definir una arquitectura de software de tres capes, presentació, negoci i dades amb unes estructura de dades pròpia definida en la base de dades del nostre servidor.

4.3.1.3. Implementació

També coneguda com fase de disseny del programa. Es realitzen els algorismes necessaris pel compliment dels requisits de l'aplicació prèviament definits. S'utilitzaran les eines descrites a la fase d'anàlisi del sistema.

Principalment s'implementarà codi font utilitzant prototips, i sistemes prova-error per corregir els bugs³⁰ del programa.

És una de les parts més costoses en recursos i temps del projecte. Encara que sembli la fase més important del cicle de vida, les dues etapes anteriors són les més rellevants. Un error en l'anàlisi pot provocar un error al disseny tècnic i posteriorment a d'implementació, és a dir, un error en cascada (com el seu nom indica) . Sempre serà més costos un error en el disseny o en l'anàlisi que un error en l'implementació.

4.3.1.4. Proves

Els elements ja programats s'uneixen per compondre el sistema i es comprova el seu correcte funcionament. Si l'aplicació compleix els requisits funcionals del SRD i no es troben bugs, aquesta podrà passar a la següent fase.

Els testers juguen un paper molt important en el cicle de creació del projecte. Com explicàvem anteriorment, els errors són en cascada igual que la seva producció. Així doncs un error en aquesta etapa suposaria tornar a alguna de les etapes anteriors.

4.3.1.5. Implantació/Manteniment

El software obtingut es posa a producció. S'implanten els nivells de software i hardware que componen el projecte. La implantació és la fase amb més duració i amb més canvis en el cicle d'elaboració del projecte.

Durant l'explotació del sistema de software poden sorgir canvis, per corregir errors o bé per introduir millores. Totes les modificacions es recolliran al Document de Canvis.

³⁰ Defecte de software (bug) és el resultat d'una errada o deficiència durant el procés de creació de software.

5. Disseny de l'aplicació










Abans de parlar sobre el disseny i l'arquitectura del nostre projecte cal que tinguem clar les especificacions amb les que haurem de treballar segons l'estudi previ del sistema que hem escollit a la fase d'anàlisi del sistema.

5.1. Estudi del sistema

Com sabreu al nostre anàlisi del sistema vam decidir treballar amb Joomla Framework. L'integració entre la nostra aplicació i el Framework tindrà èxit si ens adaptem a l'arquitectura global del sistema.

En el punt 2.2.1 hem explicat totes les característiques de Joomla!.

Recordem-les:

-  Utilitza PHP i bases de dades MySQL
-  Dues zones: Back-End i Front-End
-  Les aplicacions es divideixen en mòduls i aplicacions en diferents zones del sistema.
-  Els mòduls i els components estan desenvolupats en un model de tres capes MVC (model-vista-controlador).
-  Ofereix centenars de classes genèriques del Framework.
-  Seguretat per defecte del sistema.
-  Flexible amb llenguatges CSS, JavaScript.
-  Ofereix classes genèriques per interfícies gràfiques.
-  OpenSource, podem modificar qualsevol classe del Framework.

5.2. Disseny del sistema

Un cop sabem els requisits i restriccions que trobem al Framework de Joomla podem dissenyar la nostra aplicació seguint les especificacions.

Per començar, desenvoluparem un component Joomla! ja que ens ofereix integració amb una base de dades i recursos necessaris per a poder desenvolupar una aplicació.

Donat que la nostra aplicació serà de control econòmic i que això és un tema que no ha de ser visible per a tots els usuaris de la web, desenvoluparem un component de

Back-End, o sigui que només els administradors que tinguin cert permís donat pel Webmaster³¹ podran accedir a l'aplicació.

El nostre component tindrà una arquitectura de tres capes amb controladors, vistes i models.

Per tenir tots els detalls del disseny del sistema dividirem aquesta entrega en quatre punts de vista: lògic, processos, desenvolupament i físic.

5.2.1. Vista lògica

Des del punt de vista lògic o estructural definirem l'estructura de base de dades utilitzada per guardar tota la informació que alimenta la nostra aplicació i les definicions (noms, funcions, herències) dels controladors, vistes i models de l'arquitectura de tres capes de la nostra aplicació.






5.2.1.1. Estructura base de dades

Per a realitzar el disseny de la base de dades hem utilitzat l'eina tecnològica MySQL Workbench (punt 2.2.2.1).

Les especificacions de les bases de utilitzades a Joomla! corresponen als estàndards de MySQL.

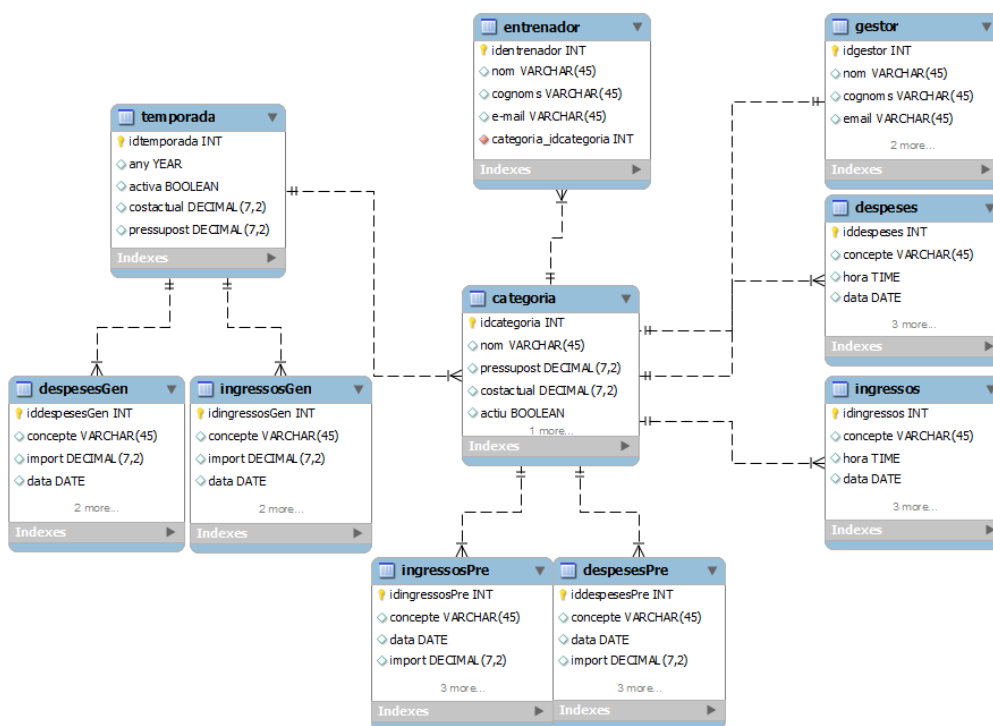
L'aplicació MySQL Workbench ens ha ajudat a dissenyar la base de dades de la nostra aplicació amb una interfície gràfica on es poden crear taules amb les seves corresponents columnes per a posteriorment guardar-hi registres.

Per a la nostra aplicació hem creat les següents taules:

-  Temporada: Guardem informació general sobre la temporada. L'any, el pressupost, el cost.
-  DespesesGen: Taula on es guarden les despeses de caràcter general. S'especifica el concepte, l'import, la data i les notes.
-  IngressosGen: Taula on es guarden els ingressos de caràcter general. S'especifiquen els mateixos paràmetres que a DespesesGen
-  Categoria: Taula on es guarden les categories actives per cada temporada. Nom, pressupost de la categoria i cost de la categoria són les taules que figuren en la taula.
-  Entrenador: Taula on guardarem l'informació general dels entrenadors de cada categoria.

³¹ Persona responsable del manteniment o programació d'un lloc web. Pot ser el responsable dels continguts del lloc o l'encarregat de l'operabilitat.

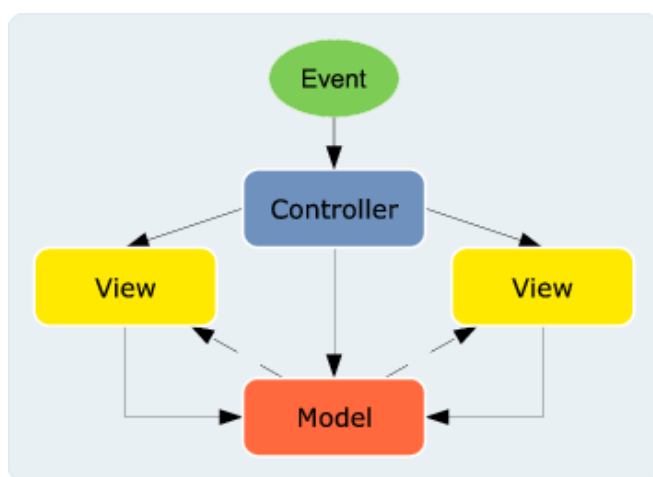
- ✚ Gestor: Taula on guardarem la informació general dels gestors/delegats de cada categoria.
- ✚ IngressosPre: Taula on guardem tots els ingressos pressupostats a inici de temporada. Aquests ingressos no són reals, són aproximacions que serviran de referència al transcurs d'una temporada. Les columnes són les mateixes que a despesesGen o ingressosGen.
- ✚ DespesesPre: Taula on guardarem les despeses pressupostades a inici de temporada. L'objectiu és igual que els ingressos pressupostats però amb despeses.
- ✚ Despeses: Taula on guardem les despeses reals al llarg d'una temporada. Guardarem informació sobre la despesa a més del nom de la persona que la ha introduït. El nom l'obtidrem segons la persona que hagi iniciat la sessió.
- ✚ Ingressos: Taula on guardarem els ingressos reals al llarg d'una temporada. La informació de la taula és igual a la de Despeses.



5.2.1.2. Definicions

Després de dissenyar una estructura de dades per a la nostra aplicació cal tenir ben clar com organitzarem la nostra aplicació. Joomla Framework ens proporciona compatibilitat amb un model de tres capes (model-vista-controlador), per tant estructurarem el nostre component amb aquesta arquitectura.

Cal tenir clar a més de les definicions de les tres capes la relació que hi ha entre elles. El controlador tindrà un model i N vistes associades a ell. Això permetrà al controlador commutar amb les vistes conegudes i obtenir les dades del seu model.



5.2.1.2.1. Controladors

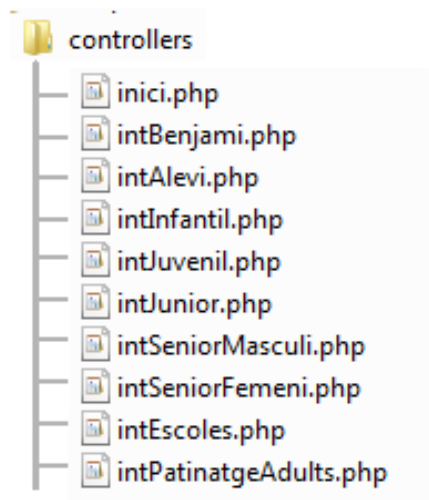
Són els components encarregats de manipular d'iteració amb l'usuari final, manipulant el model i finalment escollir la vista per mostrar-la a d'interfície d'usuari. És el responsable de la resposta a les accions que l'usuari demana a través de la vista.

En el framework utilitzat seran arxius .php on es definirà una classe que hereta de JController, una classe genèrica de Joomla!.

A l'intranet del HCR Cent Patins hem volgut tenir un espai general per a cada categoria del Club a més d'un espai d'inici on poder fer tasques de caràcter més generals (inici de temporada, finalització, resultats globals..). Aquest esquelet ens permet tenir un controlador per a cada una d'aquestes zones.

Un dels requisits funcionals de l'aplicació és que la gestió estigues dividida per categories. Amb un controlador i un model per cada "zona de categoria" podem satisfer aquest objectiu.

Els controladors són:



A cada un d'ells es repeteix el mateix patró de declaració de classe.

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
jimport('joomla.application.component.controller');

class IntranetCPControllerinici extends JController
{
}
```

Les primeres dues línies indiquen que és una zona de Back-End (impedeix el pas a persones sense permisos) i la importació de la llibreria.

Tot seguit sempre escrivim el nom de la classe amb el format: 'nomComponent'+ 'Controller'+ 'nomController'. Sempre s'ha d'especificar l'herència de JController.

Hem creat dos tipus de controladors, un d'inici (inici.php) i els controladors de cada categoria.

✚ Els mètodes del controlador inicial.

■ IntranetCPCControllerinici (JController)






- ▶ void __construct()
- ▶ void finalitzarTemp()
- ▶ void novaTemp()
- ▶ void novaTemp2()
- ▶ void novaTemp3()
- ▶ void novaTemp4()
- ▶ void inserirIngresPressupost()
- ▶ void guardarIngresPressupost()
- ▶ void tancarTemporada()
- ▶ void graficsTemporada()

Al constructor de la classe registrem totes les tasques associades als botons de la vista i els relacionem amb els mètodes següents, aquests generalment s'encarreguen de cridar al model per recollir les dades necessàries i de llençar la vista. En cas de que alguna de les carregues no tingui èxit, notificarem l'error en un missatge utilitzant un mètode de les classes que hereten de JController.

✚ Mètodes dels controladors de les categories

■ IntranetCPCControllerint+'nomCategoria' (JController)

- ▶ void display()
- ▶ void __construct()
- ▶ void vistaInserirDespesa()
- ▶ void guardarDespesa()
- ▶ void vistaInserirIngres()
- ▶ void guardarIngres()
- ▶ void vistaEditar()

-  `void editarIngres()`
-  `void editarDespesa()`
-  `void eliminar()`
-  `void actualitzarDespesesTotals()`
-  `void actualitzarIngressosTotals()`

A la funció `display()` cridem la vista per defecte, el panell de control, i la mostrem. Al constructor, al igual que al controlador d'inici, registrem les tasques dels botons. Els següents mètodes s'encarreguen o bé de carregar les vistes per inserir despeses o ingressos, o d'enviar al model les dades recollides a les vistes per a que les guardi a la base de dades.

Per a que els controladors de Joomla! puguin funcionar correctament hem de crear un dispatcher³² que commuti els diferents controladors de l'aplicació segons les peticions dels usuaris.

El dispatcher es crea com a punt de partida de l'aplicació. A la nostra aplicació el punt de partida s'anomena `admin.intranetCP.php`. Aquest arxiu conté totes les referències als altres controladors de l'aplicació. Així quan executem l'aplicació, el Framework accedeix a aquest fitxer i segons la variable global `controllerName` (extensible a tota l'aplicació) el dispatcher commuta a un controlador o a un altre. En el nostre cas inicialitzem per defecte `inici.php` i li assignem la vista per defecte `inici`. En cas de que a la pantalla d'inici polsem a la pestanya de Benjamí, el dispatcher rebrà el `controllerName = Benjamí` i executarà el controlador corresponent.

³² Part d'un programa encarregada de llançar un procés en el servidor.

En la següent imatge podem observar la part inferior de la ribbon³³ on hi ha les pestanyes amb cada una de les zones de l'aplicació.



5.2.1.2.2. Models

Són els components responsables del manteniment dels registres de la base de dades. El model s'encarrega de la representació, el càlcul de les dades i del posterior enviament d'aquestes a alguna de les vistes associades a ell i al controlador.

Als nostres models de l'aplicació hem creat funcions amb totes les consultes necessàries per a l'obtenció, manteniment i creació de dades relacionades amb cada categoria, és a dir, registres de la base de dades. Recordem: un model, un controlador i N vistes per cada categoria.

De la mateixa manera que els controladors, el Framework ens ofereix la classe JModel. A cada model de la nostra aplicació afegirem el següent format de classe:

```
<?php
defined('_JEXEC') or die();
jimport('joomla.application.component.model');

class IntranetCPModelinici extends JModel
{
}
```

Restringim l'accés, importem la llibreria del component i definim la classe com una herència de JModel. El format de nom de la classe és similar al dels controladors però amb "Model" entre el nom del component i el nom del model.

De la mateixa manera que hem creat dos tipus de controladors, també implica la construcció d'un model per a cada un d'aquests tipus.

³³ Interfície gràfica d'usuari, normalment a la part superior d'una finestra, on s'exposen totes les funcions que pot realitzar l'aplicació.

✚ Mètodes model inici (inici.php)

■ IntranetCPModelinici (JModel)

- ▶ array[string] getDadesCategoria()
- ▶ object getTemporadaActiva()
- ▶ array[string] getIngressosPressupostats()
- ▶ array[string] getCategoriesActives()
- ▶ array[string] getGraficReal()
- ▶ array[string] getGraficPressupost()
- ▶ int getAny()
- ▶ bool crearTemporada(date \$inici, date \$final, int \$any)
- ▶ bool tancarTemporada(int \$id)
- ▶ bool crearCategoria(int \$id, int \$nom)
- ▶ bool inserirDespesaPressupost(string \$concepte, date \$data, float \$import, string \$notes, string \$tipus, int \$categoria)
- ▶ bool inserirIngresPressupost(string \$concepte, date \$data, float \$import, string \$notes, string \$tipus, int \$categoria)

En el model tenim mètodes que ens retornen informació determinada de la base de dades en llistes o objectes per a que posteriorment siguin mostrats a les vistes i mètodes que ens permeten inserir registres a la base de dades. Per interactuar amb la base de dades s'utilitza la classe JFactory i operacions en llenguatge SQL.

✚ Mètodes model per categoria

■ IntranetCPModelint+Categoria

- ▶ array[string] getDespeses()
- ▶ array[string] getIngressos()
- ▶ array[string] getDespesesPressupostades()

- ▶ array[string] getIngressosPressupostats()
- ▶ bool inserirDespesa(string \$concepte, date \$data, float \$import, string \$notes, string \$usuari, float \$unitatpreu, int \$quantitat)
- ▶ bool inserirIngres(string \$concepte, date \$data, float \$import, string \$notes, string \$usuari, float \$unitatpreu, int \$quantitat)
- ▶ bool actualitzarIngresos()
- ▶ bool actualitzarDespeses()
- ▶ bool actualitzarIngres(int \$id, string \$concepte, date \$data, float \$import, string \$notes)
- ▶ bool actualitzarDespesa(int \$id, string \$concepte, date \$data, float \$import, string \$notes)
- ▶ bool eliminarDespesa(int \$id)
- ▶ bool eliminarIngres(int \$id)
- ▶ object getDespesaPerId(int \$id)
- ▶ object getIngresPerId(int \$id)
- ▶ int getTotalDespeses()
- ▶ int getTotalIngresos()

Són tots els mètodes encarregats de l'obtenció i manteniment (inserció, eliminació, modificació) de dades de cada una de les categories.

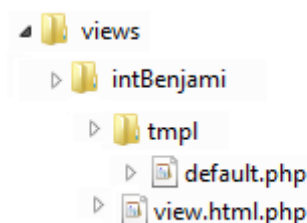
5.2.1.2.3. Vistes

Són els components encarregats de mostrar la informació al usuari, generalment per una interfície d'usuari. Poden existir múltiples vistes per model.

A Joomla Framework les vistes es divideixen en dues parts. Un arxiu .php on declarem la nostra classe que hereta de JView descriu la part lògica de la vista. En aquesta classe recollim les dades que necessitem del model, inicialitzem els objectes de l'interfície que la classe ens ofereix (botons, títols, etc) i llancem els templates.

Els templates són la segona part d'una vista. En podem tenir N subjectes a una vista. Aquests són arxius .php, encara que el seu format és en HTML, que contenen l'esquelet on posteriorment carregarem les dades que recull la classe.

Un exemple:

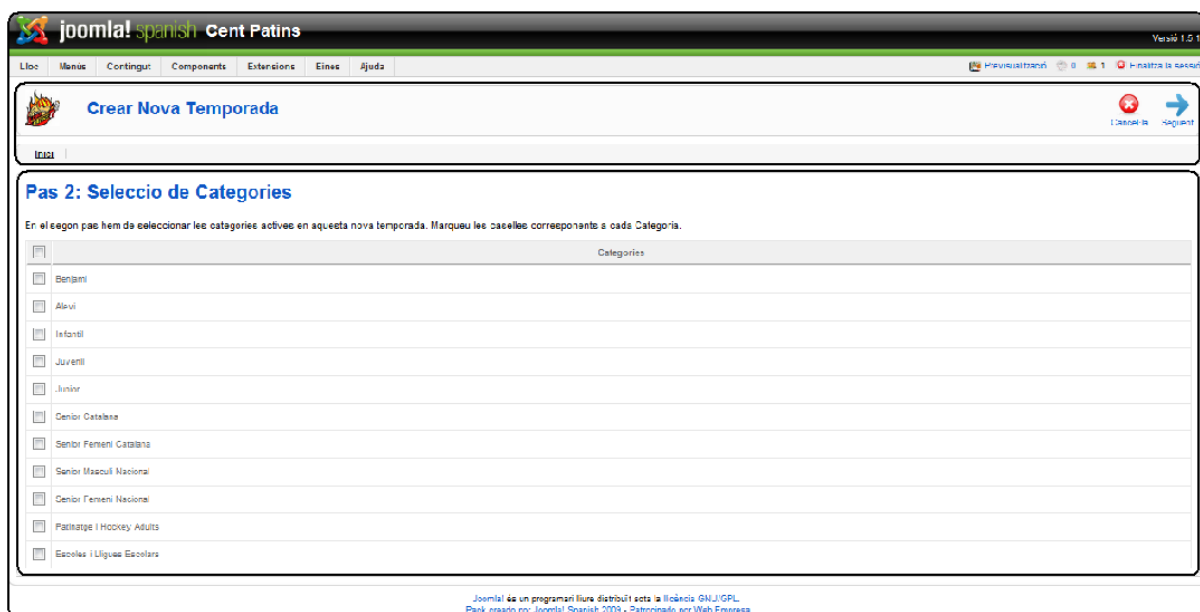


A la carpeta views de la nostra aplicació trobem la vista intBenjami que conté la classe de la vista (view.html.php) i els templates associats, en aquest cas només en tenim un que és diu default.php.

El nom del arxiu que conte la classe sempre és view.html.php. Per les vistes és important que la carpeta contenidora dels arxius tingui un nom coherent dins de la lògica del sistema. Quan llancem una vista des de un controlador, indiquem la carpeta contenidora dels arxius de la vista.

Els templates es poden anomenar com desitgem. Si el nombre amb *default*, serà el template que es llanci per defecte.

L'estructura per allotjar els nostres templates que suporta Joomla! utilitza contenidors.



Hi ha un contenidor pare que conté el menú del Back-End (menú superior horitzontal) on es poden accedir als diferents components i a la configuració global del sistema. Aquesta zona no es modificable.

El següent contenidor és la ribbon. El layout d'aquesta barra és sempre igual, a la part superior podem allotjar una imatge, un títol i els botons que desitgem. A la part inferior trobem un sub menú que ens permet navegar entre les diferents zones del component.

Els botons de la nostra aplicació són personalitzats, hem creat els botons utilitzant sentències en CSS assignant un arxiu en format png³⁴, paràmetres de disseny i el text associat.

El contenidor inferior és l'encarregat de mostrar el template que creem a cada vista. En aquesta zona podem crear tots els elements que desitgem amb HTML i donar estil amb CSS. També podem inserir petits mòduls de JQuery.

Per al control i validació de les vistes utilitzem funcions creades en JavaScript que controlen els esdeveniments de cada vista (botons, checkboxes, textboxes, etc).

A l'annex trobareu les imatges dels templates i dels botons que s'han creat per a l'aplicació.

³⁴ És un format gràfic basat en un algoritme de compressió sense pèrdua per bitmaps. Permet emmagatzemar imatges amb major profunditat i contrast.

5.2.2. Vista processos

Hem dividit la nostra aplicació en processos o cassos d'ús, fet que ens permet desenvolupar diferents funcionalitats per unitats, mantenint un ordre, planificació i control de l'aplicació.

Descrivim els processos amb diagrames de flux³⁵. Donat que les funcionalitats a cada una de les categories són les mateixes, descriurem els diagrames de flux de la zona d'inici i de la zona de categories (extensible a totes elles).

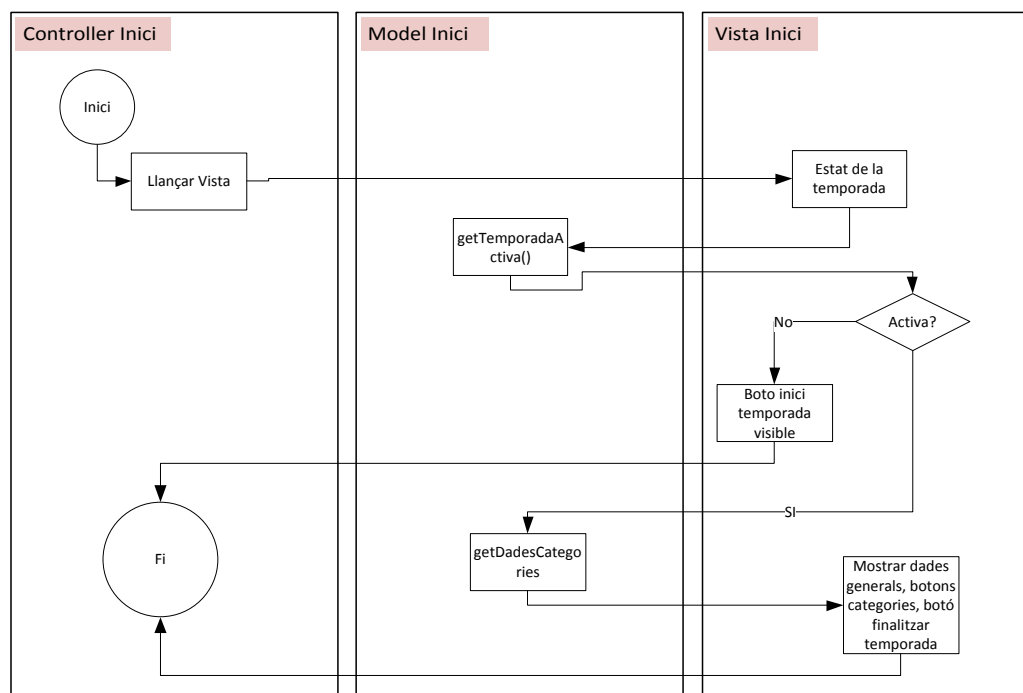
5.2.2.1. Processos zona Inici

Visualitzar panell control inici

Descripció: Aquest procés correspon a la visualització de la zona d'inici de l'aplicació. L'informació que es mostrarà serà la referent a la temporada actual amb les despeses i els ingressos reals per cada categoria i les despeses i ingressos pressupostats a principi de temporada.

Precondició: Ha fet log in al sistema.

Postcondició: Visualitza l'estat de la temporada o en el seu defecte les eines per crear-la.



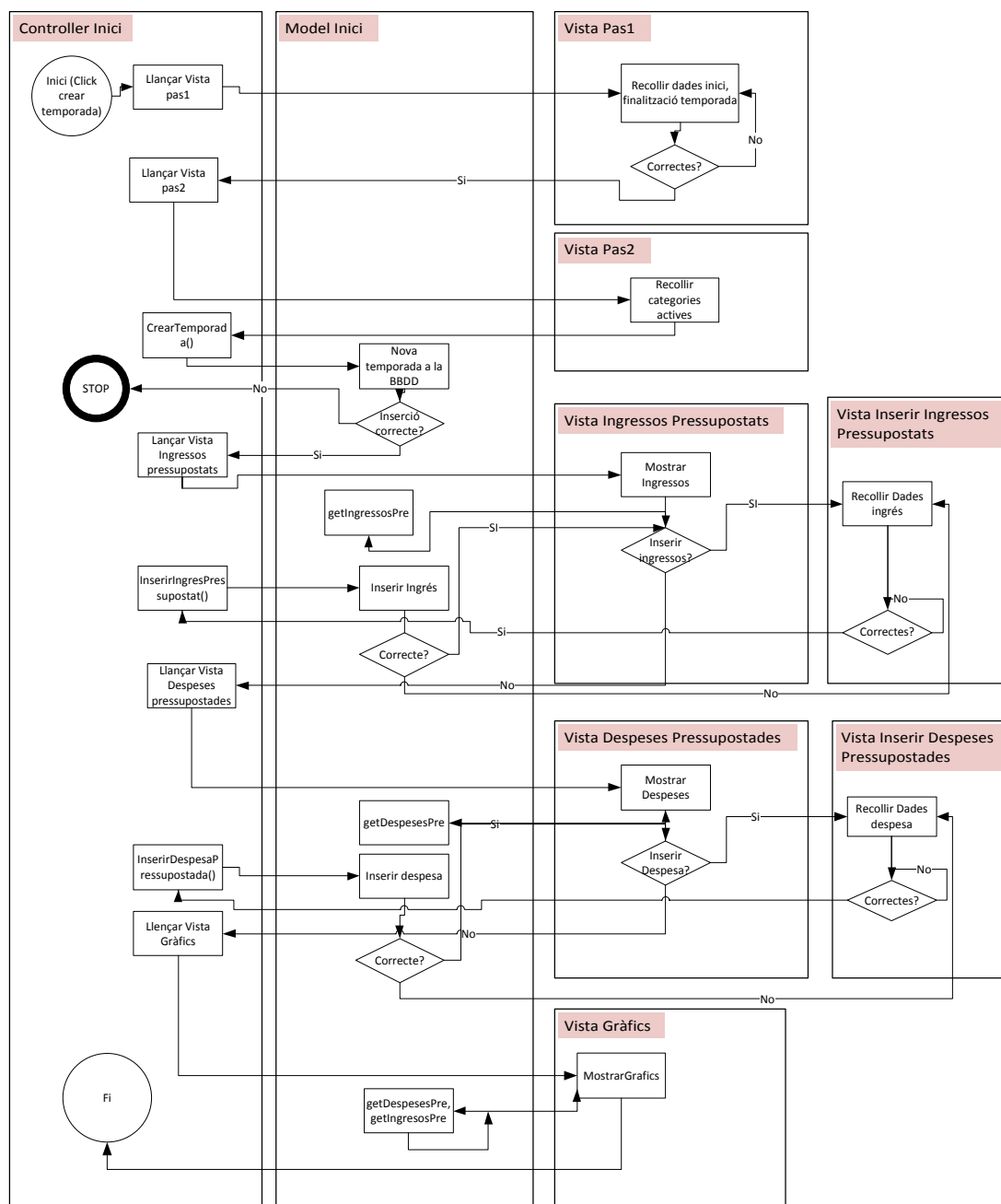
³⁵ Representació gràfica d'un algoritme o procés. S'utilitzen símbols per representar els passos de l'algoritme i representen el flux d'execució mitjançant fletxes que connecten els punts d'inici i de final.

Crear nova temporada

Descripció: Es correspon a un conjunt de petits processos que permeten crear una nova temporada a l'aplicació. El primer pas serà seleccionar una data d'inici i una de final per a la nostra temporada. Seguidament indicarem les categories que estaran actives. Els darrers passos seran introduir els pressupostos de cada categoria. S'inseriran les despeses i ingressos pressupostats per la temporada.

Precondició: Haver fet log in i no tenir cap temporada activa.

Postcondició: Temporada creada amb èxit.

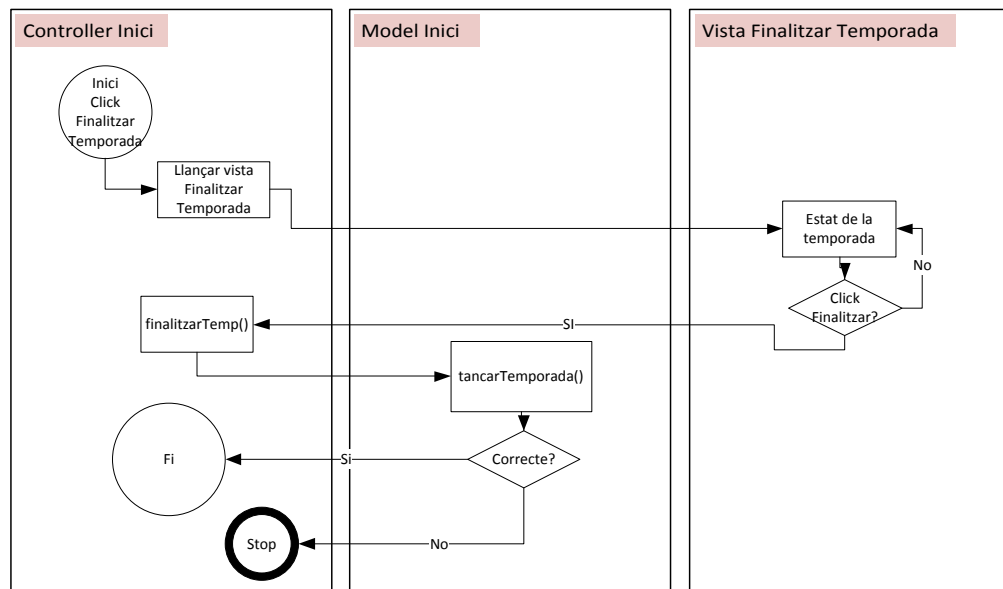


Tancar temporada

Descripció: Si per algun motiu es voles tancar la temporada per iniciar-ne una de nova hi ha l'opció de finalitzar-la. Ens mostrarà una pantalla amb les dades de la categoria i ens confirmarà l'operació.

Precondició: Haver fet log in al sistema. Tenir una temporada activa.

Postcondició: Temporada arxivada amb èxit.

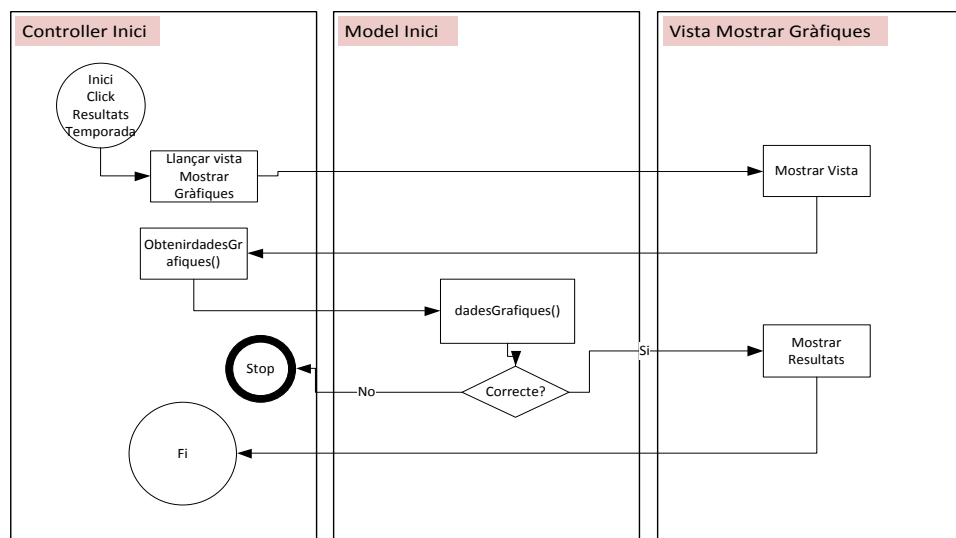


Obtenir gràfiques temporada

Descripció: Aquest cas d'ús es correspon a la mostra d'informació de la temporada en format gràfic. S'obtenen les dades de la temporada activa i es mostren en gràfics.

Precondició: Haver fet log in al sistema. Tenir una temporada activa.

Postcondició: Es mostren les gràfiques dels pressupostos i reals al l'interfície d'usuari.

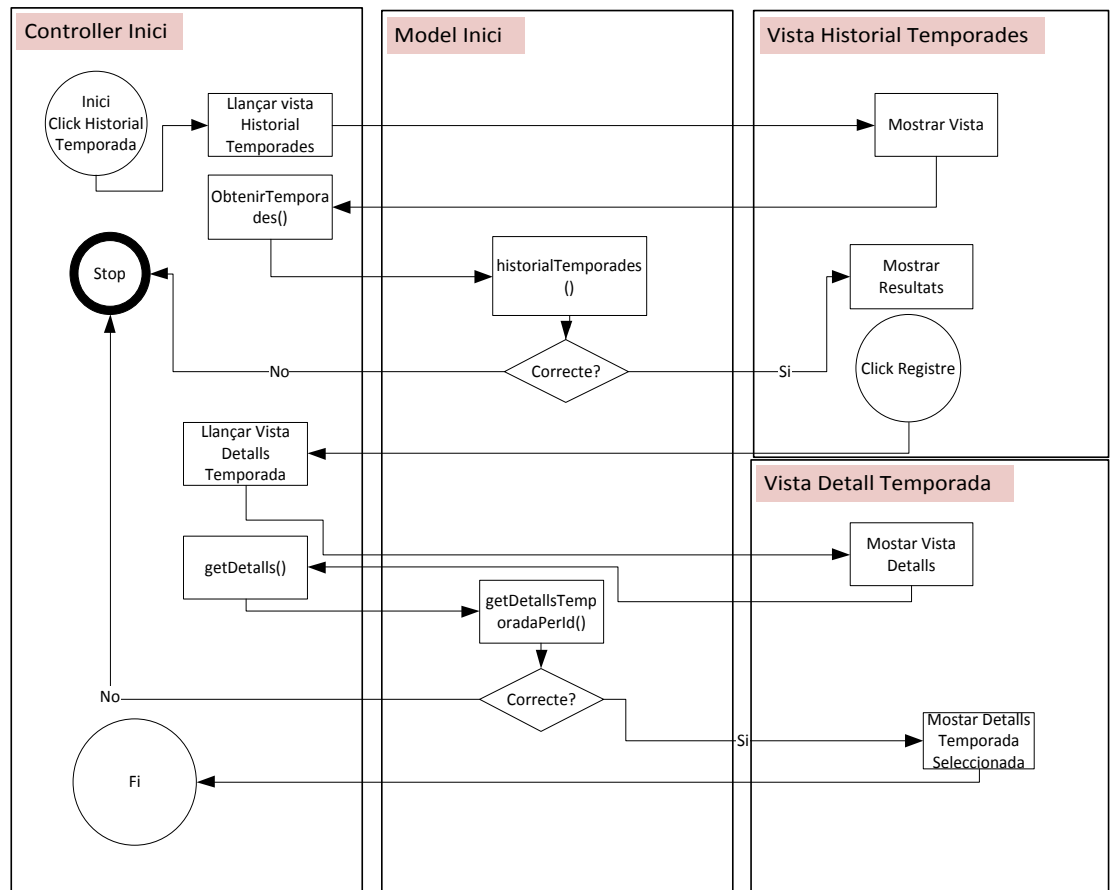


Historial temporades antigues

Descripció: Accés a la informació de despeses i ingressos d'altres temporades classificat per categories.

Precondició: Tenir alguna temporada inactiva (finalitzada).

Postcondició: Es visualitzen en primera instància totes les temporades finalitzades. Es poden veure els ingressos i despeses detalladament.



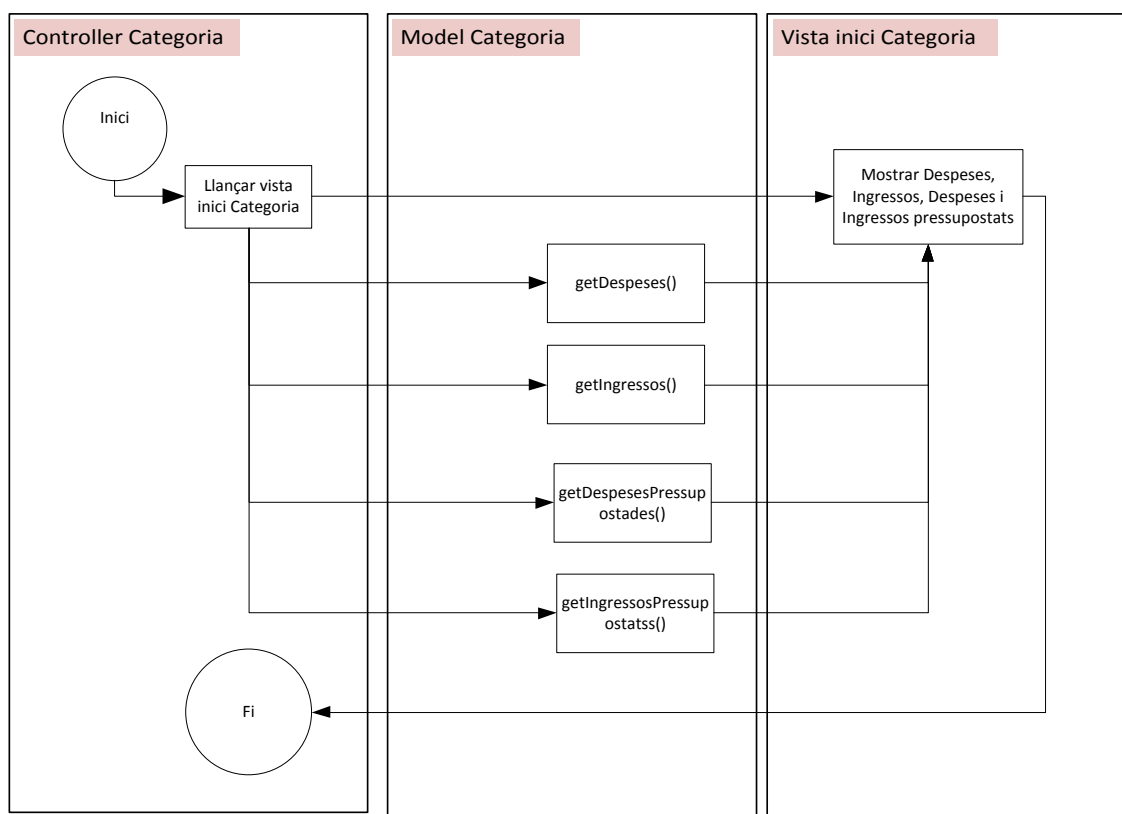
5.2.2.2. Processos zona Categoria

Visualitzar panell control categoria

Descripció: Es mostren totes les dades referents a la categoria en el mes en el que ens trobem. Es mostren tots els ingressos, despeses, ingressos pressupostats i despeses pressupostades previstos o realitzats durant el mes actual.

Precondició: Haver fet log in al sistema. Tenir una temporada activa. Tenir la categoria on es vol accedir activa.

Postcondició: Es visualitzen els ingressos i despeses pressupostats i reals del mes actual.

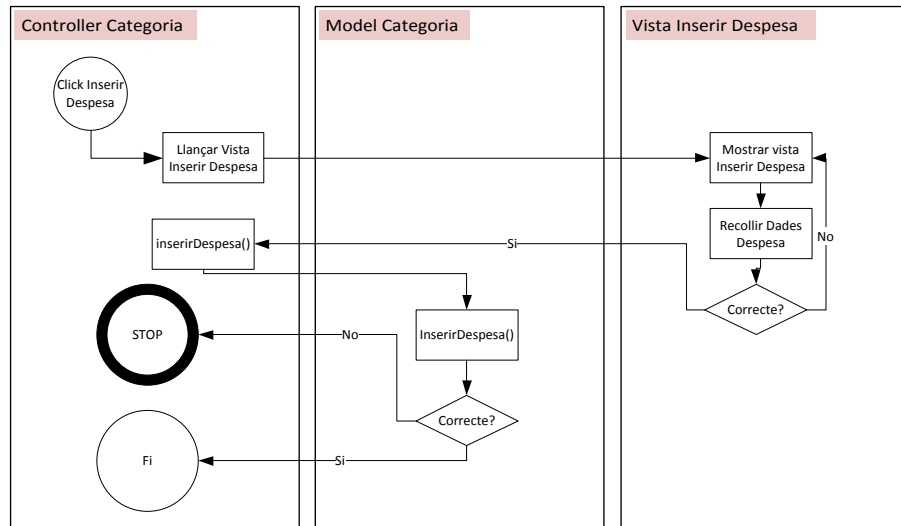


Crear nova despesa

Descripció: S'inicia una pantalla amb una plantilla que ens permet introduir les dades de la despesa detalladament. Si les dades són valides s'insereix la despesa al sistema.

Precondició: Haver fet log in al sistema. Tenir una temporada activa. Tenir la categoria on es vol inserir la despesa activa.

Postcondició: Despesa emmagatzemada amb èxit.

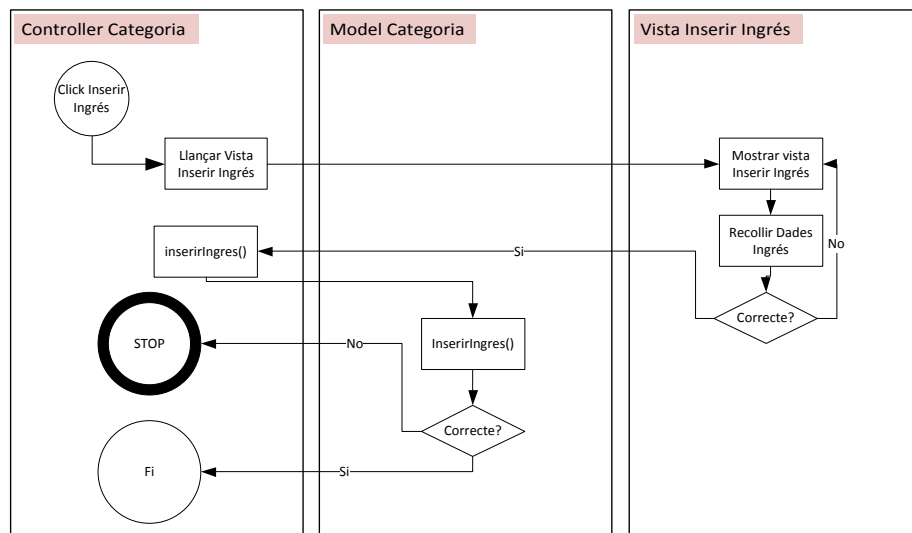


Crear nou ingrés

Descripció: S'inicia una pantalla amb una plantilla que ens permet introduir les dades de l'ingrés detalladament. Si les dades són valides s'insereix l'ingrés al sistema.

Precondició: Haver fet log in al sistema. Tenir una temporada activa. Tenir la categoria on es vol inserir el ingrés activa.

Postcondició: Ingrés emmagatzemat amb èxit.



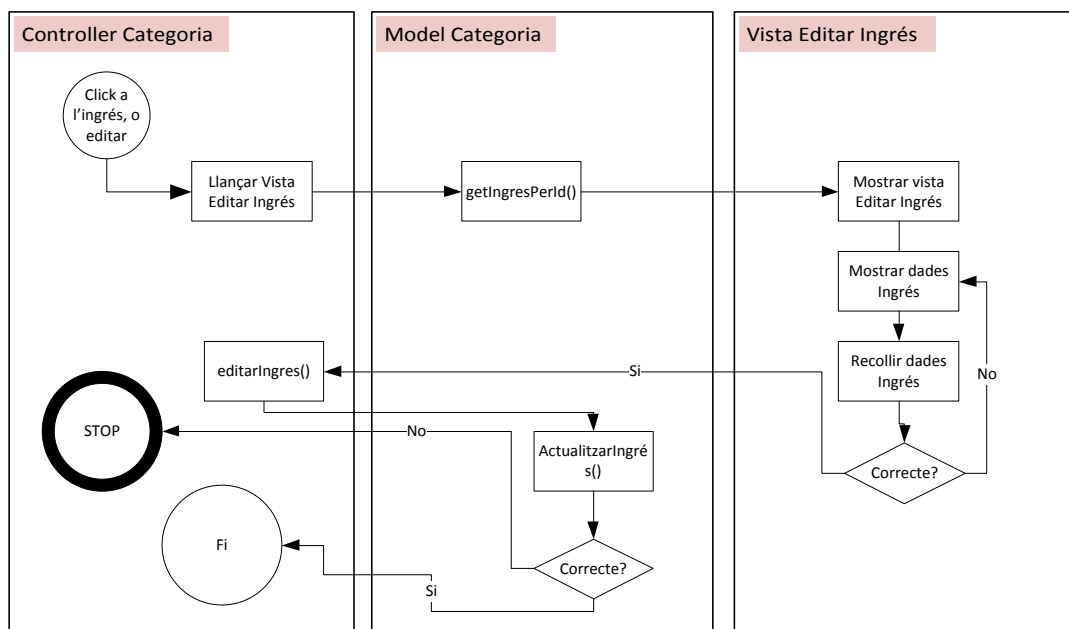
Editar ingrés

Descripció: Des de la pantalla d'inici de la categoria podem fer click sobre un ingrés per editar-lo. Es llança una plantilla amb les dades de l' ingrés. Es poden modificar les dades i si són valides s'actualitzarà l' ingrés al sistema.

Precondició: Haver fet log in al sistema. Tenir una temporada activa.

Tenir la categoria activa. Ingrés que es vol editar creat i seleccionat.

Postcondició: Ingrés editat amb èxit.

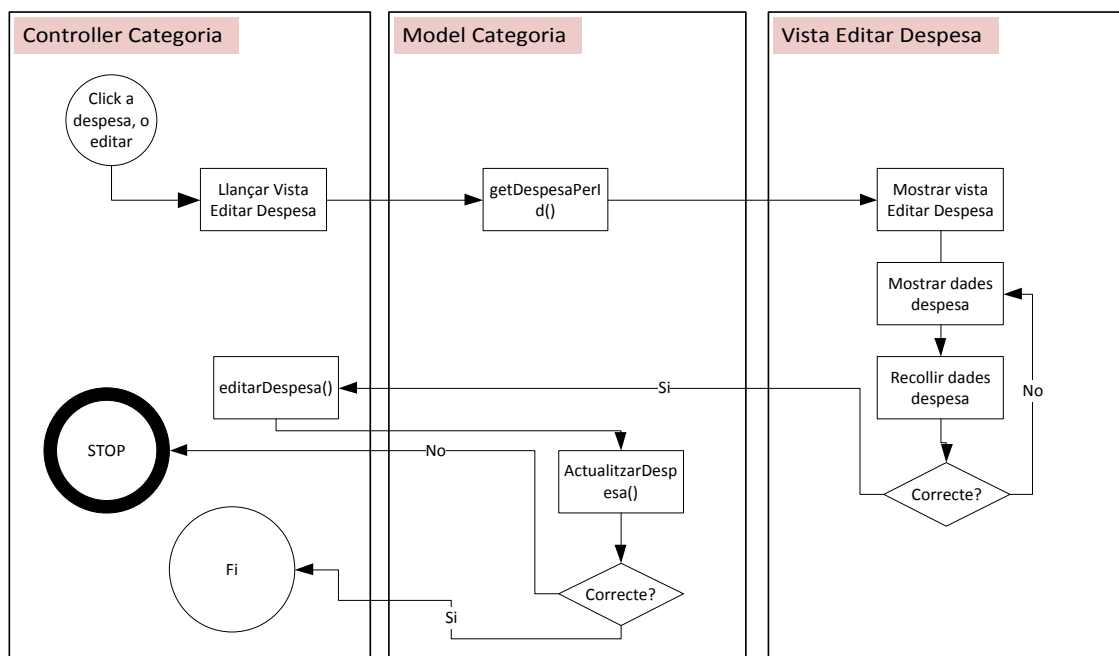


Editar despesa

Descripció: Mateix procés que amb els ingressos però es llança una plantilla per despeses. Si es valida s'actualitzarà la informació al sistema.

Precondició: Haver fet log in al sistema. Tenir una temporada activa. Tenir la categoria activa. Despesa que es vol editar creada i seleccionada.

Postcondició: Despesa editada amb èxit.



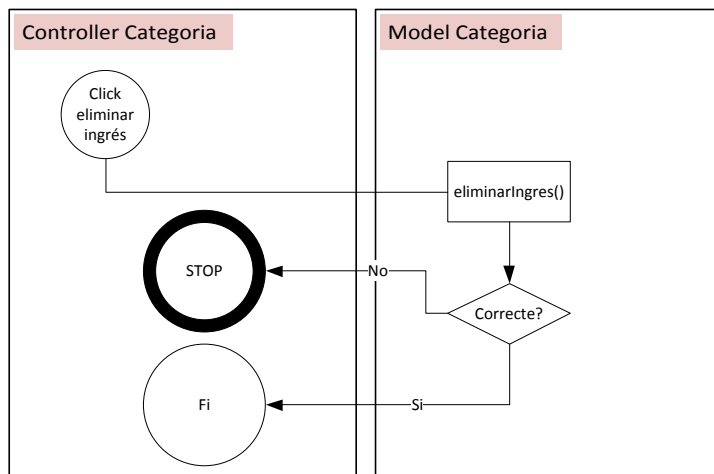
Eliminar ingrés

Descripció: Quan es fa click sobre un ingrés i s'associa la tasca d'eliminar, s'envia la petició i quan el sistema ho realitza es mostra un missatge certificant que s'ha eliminat amb èxit.

Precondició: Haver fet log in al sistema. Tenir una temporada activa.

Tenir la categoria activa. Ingrés que es vol eliminar creat i seleccionat.

Postcondició: Ingrés eliminat amb èxit.



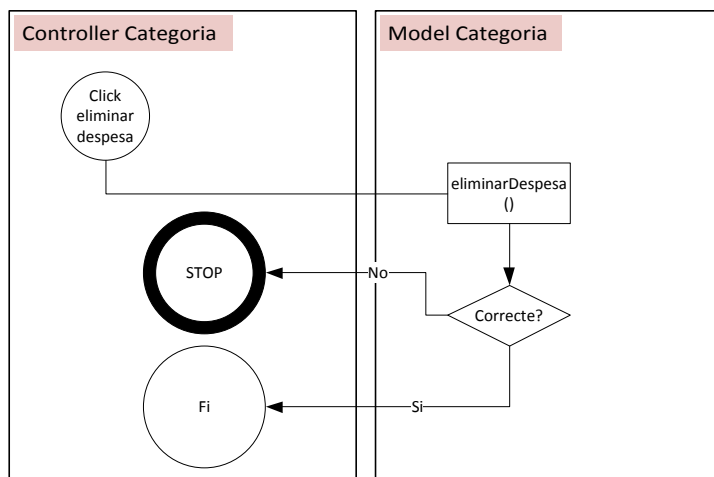
Eliminar despesa

Descripció: Quan es fa click sobre una despesa i s'associa la tasca d'eliminar, s'envia la petició i quan el sistema ho realitza es mostra un missatge certificant que s'ha eliminat amb èxit.

Precondició: Haver fet log in al sistema. Tenir una temporada activa.

Tenir la categoria activa. Despesa que es vol eliminar creada i seleccionada.

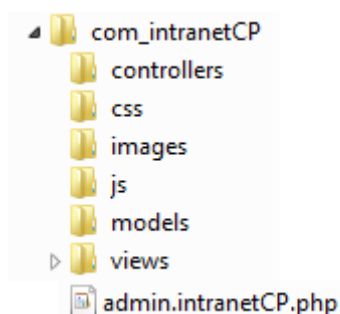
Postcondició: Despesa eliminada amb èxit.



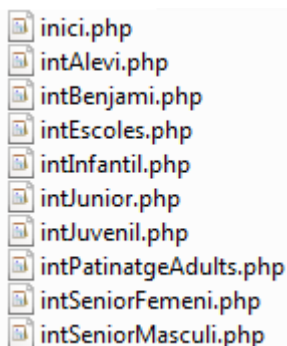
5.2.3. Vista desenvolupament

El projecte s'organitza en carpetes per facilitar-ne el desenvolupament. Aquesta estructura és la forma estàndard de qualsevol aplicació Joomla! desenvolupada amb l'arquitectura MVC.

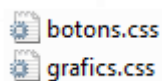
Vista global:



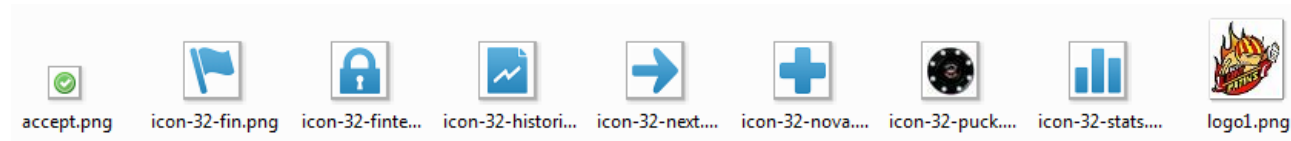
Carpeta controllers:



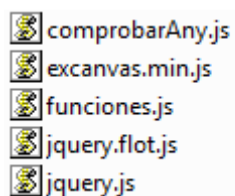
Carpeta css:



Carpeta images:

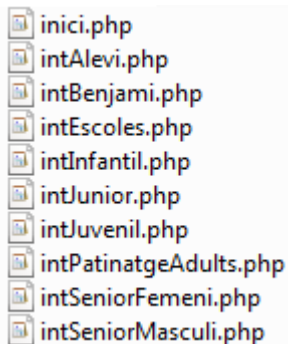


Carpeta js:



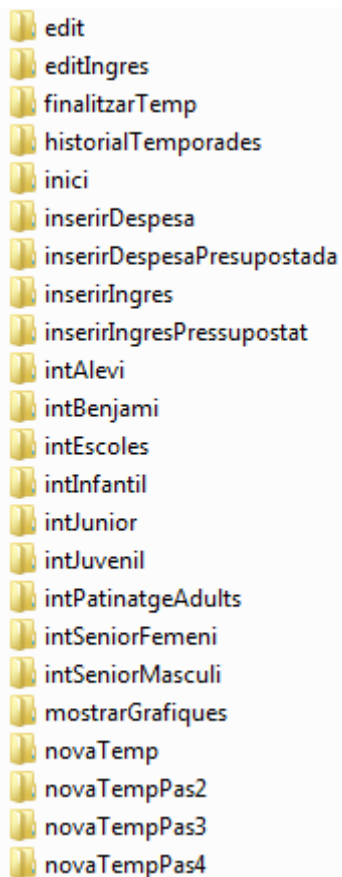
- comprobarAny.js
- excanvas.min.js
- funciones.js
- jquery.flot.js
- jquery.js

Carpeta models:



- inici.php
- intAlevi.php
- intBenjami.php
- intEscoles.php
- intInfantil.php
- intJunior.php
- intJuvenil.php
- intPatinatgeAdults.php
- intSeniorFemeni.php
- intSeniorMasculi.php

Carpeta views:



- edit
- editIngres
- finalitzarTemp
- historialTemporades
- inici
- inserirDespesa
- inserirDespesaPresupostada
- inserirIngres
- inserirIngresPressupostat
- intAlevi
- intBenjami
- intEscoles
- intInfantil
- intJunior
- intJuvenil
- intPatinatgeAdults
- intSeniorFemeni
- intSeniorMasculi
- mostrarGrafiques
- novaTemp
- novaTempPas2
- novaTempPas3
- novaTempPas4

Cada una d'aquestes carpetes conté l'arxiu de la classe (view.html.php) i el template (default.php)

5.3. Conclusions del disseny

L'adopció del model distribuït de tres capes de Joomla! ens ha permès, com a característica principal, la reusabilitat de codi. Això suposa un benefici en molts aspectes del projecte, redueix la quantitat de recursos i els costos de desenvolupament i manteniment.

Tanmateix aquesta arquitectura de disseny ens permet la funcionalitat que requereixen aplicacions d'aquest tipus, seguretat, rapidesa en el flux d'execució i sobretot manteniment i flexibilitat.

La flexibilitat del sistema és molt superior a una estructura d'una capa o lineal. Podem desenvolupar futurs processos (casos d'ús) sense necessitat de modificar els existents. Clarament Joomla Framework ens ha donat un cop de mà i a pesar de que l'aprenentatge de l'arquitectura i de les classes ha suposat un retràs en la planificació del projecte, a llarg termini podem estar satisfets dels resultats.

6. Tests

Les proves de software són els processos que permeten verificar i revelar la qualitat d'un producte software. Les hem utilitzat per identificar possibles errades d'implementació, qualitat o usabilitat.

En aquesta fase del cicle de desenvolupament del software provarem els casos d'ús construïts per comprovar el grau de compliment respecte les especificacions inicials (requisits funcionals i no funcionals) del nostre sistema.

Hem de distingir entre els bugs i els defectes de forma. En un defecte de forma el programa realitza alguna cosa que l'usuari no espera. Un bug és un error de programació, una errada en la semàntica. També pot conduir a tenir un defecte de forma.

En la nostra aplicació, nosaltres mateixos hem testejat el software de manera iterativa. Des de el principi del desenvolupament s'han anat fent les comprovacions i les pertinents modificacions als bugs o errors de forma detectats.

Per a cada un dels requisits que s'especifiquen a la primera part del projecte, expliquem quines proves hem realitzat i analitzem els resultats obtinguts. Principalment hem realitzat proves a les interfícies d'usuari. Entenem que els defectes de forma s'han anat solucionant al llarg del desenvolupament.

Requisits funcionals:

- ✚ Altes, baixes, modificacions de despeses per categoria.

Hem escollit una categoria activa en una temporada activa i hem intentat l'inserció de despeses. Si ens hem deixat algun camp no ens ha deixat continuar, mostrant un missatge d'error indicant el paràmetre erroni. No es pot continuar sense tenir les dades correctes. Hem intentat marxar de la pantalla mitjançant els menús superiors (entorn Joomla). No ha estat possible ja que estan deshabilitats. Només es possible guardar o cancel·lar.

Al editar un registre ens apareixen tots els camps omplerts amb l'última informació emmagatzemada. Si esborrem algun camp i desem ens mostra un missatge d'error indicant el camp erroni. Si introduïm hores o dates incorrectes es mostra un missatge d'error. No ens permet introduir imports negatius.

Hem intentat eliminar sense seleccionar cap registre. Ens mostra un missatge d'error. Si esborrem més d'una despesa el programa reacciona correctament esborrant tots els seleccionats.

✚ Altes, baixes, modificacions d'ingressos per categoria

Les validacions són iguals que les de despeses. L'aplicació respon correctament davant d'algun paràmetre incorrecte en l' inserció o edició. Múltiple esborrat correcte.

✚ Control d'accés dels usuaris de l'aplicació.

A la part d'administració de Joomla hem intentat entrar amb usuaris desconeguts pel sistema, no ha estat possible l'entrada al Back-End. Hem introduït la direcció de la nostra aplicació a la barra de l'explorador sense tenir cap sessió activa, ens ha demanat autenticació.

✚ Mostra de dades de caràcter econòmic per categories.

Dades mostrades correctament.

✚ Gràfics estadístics de l'estat de comptes.

Al entrar a Resultats Temporada des de el panell de control d'inici ens apareixen dos gràfics. Un amb les despeses i ingressos reals i l'altre pressupostats. Hem intentat visualitzar el gràfic sense despeses ni ingressos, no es mostra cap línia.

✚ Informes detallats de l'estat de comptes anuals.

En fase de desenvolupament. Fora del termini previst.

✚ Arxivador de temporades.

Al finalitzar una temporada observem que tenim la temporada a l'historial de temporades amb els seus registres. Sempre es guarden les temporades que tanquem, encara que no tinguin registres associats.

✚ Creació de pressupostos detallats amb despeses i ingressos.

Al pas 1 de crear una nova temporada hem intentat inserir dates errònies o mesos incoherents, ens mostra un missatge d'error i no ens permet continuar. Al següent pas ens demana les categories, només podem cancel·lar o continuar. No ens deixa continuar sense seleccionar cap categoria.

A l' inserció de despeses i ingressos pressupostats hi ha controls de validació de l' informació. No ens permet introduir dades incorrectes o deixar camps sense especificar, ens apareixen missatges d'error. Seguim sense poder abandonar el procés per altres botons que no sigui el de cancel·lar.

✚ Flexibilitat per a inserir noves funcionalitats.

No es poden realitzar tests a d'interfície.

✚ Còpies de seguretat i recuperació de dades.

No es poden realitzar tests a d'interfície.

7. Conclusions

L'estudi del projecte Intranet pel HCR Cent Patins (les eines, les tecnologies, la metodologia, l'arquitectura, etc) ens ha permès entendre molts conceptes aplicables en molts àmbits, especialment el professional.

Fem ressò de l'evolució constant del projecte des de l'inici. L'aprenentatge de noves arquitectures sumat al perfeccionament de tecnologies conegudes ha fet que el projecte cada dia estigués més estructurat i fos més coherent.

Si comparem l'aplicació quan s'havien implementat dos casos d'ús amb l'actual ens adonem de la millora de l'organització lògica dels processos. Nosaltres avaluem aquesta transformació positivament encara que la planificació del projecte estimada en les primeres fases del cicle no s'ha complert. El canvi que ha experimentat el software en la fase de desenvolupament ens ha fet tornar, en molts casos, enrere a fases de disseny per redistribuir estructures.

Gran part de la desviació es deu a la complexitat de l'arquitectura del marc de treball del projecte. Ens ha portat més hores de feina de les que teníem planificades estructurar una aplicació amb les arquitectures requerides. El poc coneixement d'aquests patrons ha propiciat errades de disseny i defectes de forma.

Valorem el progrés positivament, possiblement les desviacions ens han ajudat a comprendre el sistema més clarament.

Aquestes errades comeses en la planificació i en les fases d'anàlisi i disseny ens ajudaran, segurament, a millorar els nostres futurs treballs tant acadèmics com professionals.

Creiem que Joomla! ens ha ajudat a aconseguir, amb major o menor grau, els set requisits de qualitat del software: funcionalitat, confiabilitat, usabilitat, eficiència, mantenibilitat, portabilitat i escalabilitat. No obstant, seguirem treballant en millorar l'aplicació i en incorporar noves ampliacions a mesura que el client les estimi oportunes.

7.1. Futures modificacions i ampliacions

Segur que al llarg de la vida de la nostra aplicació apareixeran noves perspectives, noves inquietuds o simplement nous requeriments que ens plantejaran canvis i millores. Afortunadament, l'arquitectura utilitzada en aquest projecte garanteix gran escalabilitat i flexibilitat a l'hora d'inserir noves ampliacions.

Com a futures ampliacions ens marquem objectius com per exemple la creació d'informes en PDF o fulles d'Excel. Un altre ampliació interessant seria la de permetre treballar amb l'aplicació off-line, és a dir sense accés a la xarxa, i posteriorment sincronitzar tots els canvis amb el servidor.

No tenim unes ampliacions ben definides, però estem oberts a qualsevol suggerència per part del client.

7.2. Valoració personal

A nivell personal m'agradaria destacar tot el coneixement que he adquirit durant el projecte. Tecnològicament he copsat arquitectures i conceptes completament desconeguts fins ara, que m'han ajudat moltíssim en el desenvolupament del projecte i de ben segur que en futurs projectes.

D'altra banda, he entès i he experimentat les fases del cicle de vida del software. Les coneixia i pensava entendre-les, però realment fins que no formes part d'un projecte no sents l'importància de cada una de les seves fases.









He analitzat, dissenyat i desenvolupat amb deficiències a l'inici del projecte, però el procés recursiu fins a trobar un disseny coherent m'ha ensenyat a tractar el software de manera diferent. Podria dir que aquesta ha sigut la meva primera experiència real en la creació d'aplicacions i n'estic molt satisfet per l'evolució tecnològica, metòdica i personal.

Em sento molt més preparat per introduir-me al món laboral. Fa uns mesos pensava que mai podria arribar a tenir un coneixement extens en aquesta disciplina per la manca d'hores i l'inseguretat personal respecte la matèria. No obstant, el projecte m'ha donat una empenta i m'ha demostrat que puc ser capaç d'aprendre i utilitzar noves tecnologies i noves arquitectures ràpidament.







També he après a organitzar-me millor i a prioritzar les tasques.

Referències

Bibliografia

-  Graf, Hagen (2006). *Building Websites with Joomla*. Birmingham: Packt Publishing.
-  Graf, Hagen (2007). *Building Websites with Joomla 1.5 Beta 1*. Birmingham: Packt Publishing.
-  Kennard, James (2007). *Mastering Joomla! 1.5 Extension and Framework Development*. Birmingham: Packt Publishing.
-  Dawson, Brandon i Canavan, Tom. *Joomla! Cash*. Birmingham: Packt Publishing.
-  Trevejo, Juan Antonio (2007). *Joomla! para principiantes*.
-  Blakeley, Tessa (2007). *Joomla! Template Design*. Birmingham: Packt Publishing.
-  LeBlanc, Joseph (2007). *Learning Joomla! 1.5 Extension Development*. Birmingham: Packt Publishing.
-  Rahmel, Dan (2007). *Professional Joomla*. Indianapolis: Wiley Publishing.

Webgrafia

-  Portal Joomla Español
<http://www.joomlaspanish.org/>
-  Comunitat Joomla
<http://www.joomla.org/>
-  Documentació general Joomla!
http://docs.joomla.org/Main_Page
-  Pàgina Cent Patins
<http://www.centpatins.com>
-  Classes i mètodes
http://api.joomla.org/li_Joomla-Framework.html
<http://docs.joomla.org/Framework>
http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos
-  Arquitectura MVC.
http://docs.joomla.org/Tutorial:Developing_a_Model-View-Controller_Component_-_Part_1

http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

<http://downloads.joomlacode.org/frsrelease/3/7/6/37642/JoomlaSpanish.pdf>

<http://www.phpclasses.org/package/5935-PHP-Base-MVC-classes-for-developing-Joomla-extension.html>

<http://matware.com.ar/joomla/como-crear-componentes-mvc-para-joomla-1-5.html>

Creació Components Joomla!

<http://www.nosolocodigo.com/componentes-para-joomla-15-i>

Edició de pàgines de contingut estàtic Joomla!

<http://www.edujoomla.es/manuales-joomla-15/gestion-contenido/125-edicion-contenido-estatico>

JavaScript

<http://www.nosolocodigo.com/agregar-javascript-y-css-a-una-vista-en-joomla-15>

<http://www.nosolocodigo.com/usando-ajax-en-componentes-de-joomla>

CSS

http://docs.joomla.org/How_to_create_a_custom_button

<http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>

<http://matware.com.ar/joomla/creacion-de-templates-en-joomla.html>

JQuery

<http://jquery.com/>

<http://www.xploit29.com/2008/10/crear-tabs-para-cualquier-sitio-con-jquery-parte-ii/>

<http://nuevascreaciones.com/archives/153>

Validació formularis

http://docs.joomla.org/Form_validation

Expressions regulars

http://doc.async.com.br/formcheck_re/overview.html

http://gollum.inforg.uniovi.es/aii/valida_regexp.php

XAMPP

<http://www.apachefriends.org/es/xampp.html>

<http://www.joomlapanel.com/joomla-article/joomla-tutorial/86-install-joomla-local-host-server.html>

PHP

<http://www.php.net/>

<http://es.wikipedia.org/wiki/PHP>



MySQL

<http://es.wikipedia.org/wiki/MySQL>

<http://www.mysql.com/>

<http://dev.mysql.com/doc/>

<http://es.wikipedia.org/wiki/SQL>



MySQL Workbench

<http://wb.mysql.com/>



Metodologia en cascada

http://es.wikipedia.org/wiki/Desarrollo_en_cascada



Pàgina oficial W3C

<http://www.w3c.es/>

ANNEX

Vistes i imatges aplicació



Intranet Gestió HCR Cent Patins

Finalitzar Temporada
 Historial Temporades
 Resultats Temporada

[Inici](#) |
 [Escoles](#) |
 [Patinatge Adults](#) |
 [Senior Femeni](#) |
 [Senior Masculi](#) |
 [Junior](#) |
 [Juvenil](#) |
 [Infantil](#) |
 [Alevi](#) |
 [Benjami](#)

Benvolgut/da Albert Margarit

Categoria	Previsió Anual	Despeses Actuals	Estat
Escoles	487.50€	38.00€	OK
Adults	487.50€	38.00€	OK
Senior Femeni	776.00€	100.00€	OK
Senior Masculi	15129.00€	144.00€	OK
Junior	487.50€	144.00€	OK
Juvenil	100.00€	144.00€	ALERTA!
Infantil	487.50€	144.00€	OK
Alevi	15129.00€	1776.00€	ALERTA!
Benjami	100.00€	144.00€	OK

Categoria	Ingressos Actuals	Despeses Actuals	+/-
Escoles	156.00€	38.00€	156€
Adults	20.00€	38.00€	-18€
Senior Femeni	487.50€	100.00€	387.5€
Senior Masculi	280.00€	144.00€	136€
Junior	300.00€	144.00€	156€
Juvenil	100.00€	144.00€	-44€
Infantil	100.00€	144.00€	-44€
Alevi	200.00€	1776.00€	-1576€
Benjami	15129.00€	144.00€	14985€

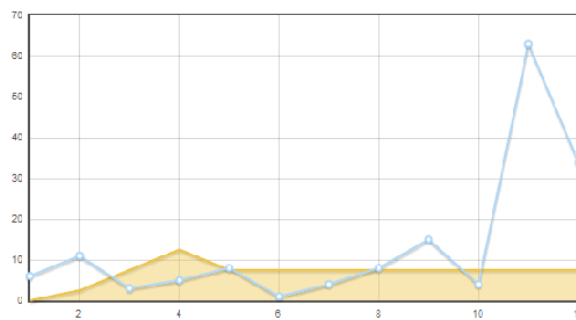


Gràfics Temporada Actual

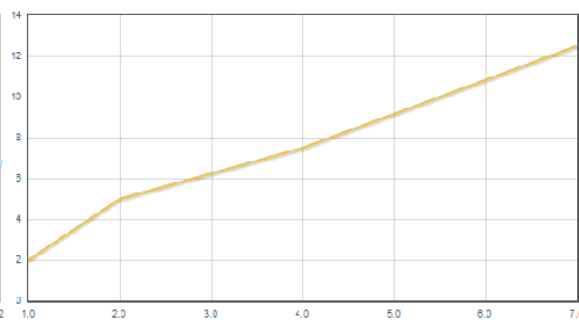


[Inici](#) | [Escoles](#) | [Patinatge Adults](#) | [Senior Femení](#) | [Senior Masculí](#) | [Junior](#) | [Juvenil](#) | [Infantil](#) | [Aleví](#) | [Benjamí](#)

Gràfic Pressupostos



Gràfic Real



Llegenda

Ingressos
Despeses



Control Aleví



[Inici](#) | [Senior Masculí](#) | [Junior](#) | [Juvenil](#) | [Infantil](#) | [Aleví](#) | [Benjamí](#)

i Despeses introduïdes amb èxit

Control Setembre 14, Setembre de 2010

Ingressos Pressupostats de Setembre

Concepte	Notes	Quantitat	Preu Unitat	Data	Import
Quotes	Quotes Alevins	1	400.00€	2010-09-01	400.00€
Quotes	Matrícules Alevins	1	1000.00€	2010-09-01	1000.00€
Subvenció	Ajuntament de Rubí	1	1000.00€	2010-01-01	1000.00€
Altres	Aportació pares Aleví	1	400.00€	2010-09-01	400.00€
Subvenció	Ajuntament de Rubí	1	1000.00€	2010-09-01	1000.00€

Despeses Pressupostades de Setembre

Concepte	Notes	Quantitat	Preu Unitat	Data	Import
Altres	Entrenadors Aleví	1	220.00€	2010-09-01	220.00€
Altres	Material i aigua	1	100.00€	2010-09-01	100.00€

Ingressos de Setembre

Concepte	Notes	Quantitat	Preu Unitat	Hora	Data	Import	Usuari
<input type="checkbox"/> Quotes	Matrícules Alevins	1	750.00€	11:38:39	2010-09-14	750.00€	Albert Margarit
<input type="checkbox"/> Quotes	Quotes Alevins	1	487.50€	11:38:05	2010-09-14	487.50€	Albert Margarit
<input type="checkbox"/> Subvenció	Ajuntament de Rubí	1	1000.00€	11:37:35	2010-09-14	1000.00€	Albert Margarit

Despeses de Setembre

Concepte	Notes	Quantitat	Preu Unitat	Hora	Data	Import	Usuari
<input type="checkbox"/> Arbitratges	Arbitratges Setembre	4	50.00	11:39:02	2010-09-14	200.00€	Albert Margarit
<input type="checkbox"/> Entrenadors	Entrenadors Aleví	1	220.00	11:37:04	2010-09-14	220.00€	Albert Margarit
<input type="checkbox"/> Material	Aigua	4	2.00	11:36:45	2010-09-14	8.00€	Albert Margarit



Introduir Despesa



Concepte: Notes: Quantitat: Preu Unitat: Hora: Data: Import: €

- Entrenadors
- seleccions
- Material
- Arbitratges
- Entrenadors
- Desplaçaments
- Ajudaments
- Federació
- Enfermeria
- Extra

Joemla és un programari lliure distribuït sota la llicència GNU/GPL.
Pack creat per Joemla Spanish 2015 - Patrocinat per Web Empresa



Introduir Ingres



Desa

Cancel·la

Concepte	Notes	Quantitat	Preu Unitat	Hora	Data	Import
Sponsors	Pedas Immobiliària	1	2000	17:50:15	2010-0-12	2000 €



Editar Despesa



Desa

Cancel·la

Concepte	Notes	Quantitat	Preu Unitat	Hora	Data	Import
Enfermeria	Lesió Carlos Genito	1	250	10:15:00	2010-00-11	250 €



Editar Ingres



Desa

Cancel·la

Concepte	Notes	Quantitat	Preu Unitat	Hora	Data	Import
Subvencio	Alevi	1	250	10:15:00	2010-00-11	250 €



Finalitzar Temporada



Confirmació Finalitzar Temporada

Cancel·la

Inici | Escoles | Patinatge Adults | Senior Femení | Senior Masculí | Junior | Juvenil | Infantil | Alevi | Benjamí

Any	Pressupost	Cost Total	Ingressos Totals	Data Inici Temporada
2015	1770.00€	15129.00€	280.00€	2015-01-01



Crear Nova Temporada



Cancel·la

Següent

Pas 1: Informació General

En aquest primer pas seleccionarem la informació General de la Temporada.

	Any	Mes
Introdueix la data d'inici	2010	0
Introdueix la data Final	2011	6



Crear Nova Temporada



Cancel·la

Següent

Pas 2: Selecció de Categories

En el segon pas hem de seleccionar les categories actives en aquesta nova temporada. Marquem les caselles corresponents a cada Categoria.

<input type="checkbox"/>	Categories
<input checked="" type="checkbox"/>	Benjamí
<input type="checkbox"/>	Alevi
<input checked="" type="checkbox"/>	Infantil
<input type="checkbox"/>	Juvenil
<input checked="" type="checkbox"/>	Junior
<input type="checkbox"/>	Senior Masculí
<input checked="" type="checkbox"/>	Senior Femení
<input type="checkbox"/>	Patinatge i Hockey Adults
<input type="checkbox"/>	Femmes i Joves Femelles



Crear Nova Temporada

Inserir Ingres Pressupost Cancel·la Següent

Ingres introduït amb èxit

Pas 3: Pressupost

Definir Ingressos Totals.

SENIOR FEMENI	SENIOR MASCULI	JUNIOR	JUVENIL	INFANTIL	ALEVI	BENJAMI
Concepte	Data	Import	Notes	Tipus		
Cuotes	2010-12-01	200.00		0		



Crear Nova Temporada

Inserir Despesa Pressupost Cancel·la Següent

Pas 3: Pressupost

Definir Despeses Totals.

SENIOR MASCULI	JUNIOR	JUVENIL	ALEVI	BENJAMI
Concepte	Data	Import	Notes	Tipus



Introduir Ingres Pressupost

Desa Cancel·la

Inici | Junior | Juvenil | Infantil | Alevi | Benjamí

Concepte	Mes	Any	Import	Notes	Tipus	Categoria
Subscripció	1	2010	1000	Ajuntament	Maneja	Alevi
2						
3						
4						
5						
6						
7						
8						
9						

Joemil és un programari lliure distribuït sota la llicència GNU/GPL.
Pack creat per Joemil Spanish 2008 - Patrocinat per Web Empresa



Introduir Despesa Pressupost

Desa Cancel·la

Inici | Patinatge Adults | Senior Femení | Senior Masculí | Junior | Alevi | Benjamí

Concepte	Mes	Any	Import	Notes	Tipus	Categoria
- seleccions -	11	2010	500	Fites i Revisions	Planu	Adults
						Adults
						Senior Femení
						Senior Masculí
						Junior
						Alevi
						Benjamí

Joemil és un programari lliure distribuït sota la llicència GNU/GPL.
Pack creat per Joemil Spanish 2008 - Patrocinat per Web Empresa

